

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Využití gamesourcingu a kolektivní
inteligence v počítačových hrách při
řešení úloh**

**Computer Games, Gamesourcing and
Swarm Intelligence in Problem Solution**

2017

Bc. Jiří Arleth

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student:

Jiří Arleth

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Využití gamesourcingu a kolektivní inteligence v počítačových hrách při řešení úloh
Computer Games, Gamesourcing and Swarm Intelligence in Problem Solution

Jazyk vypracování:

čeština

Zásady pro vypracování:

Gamesourcing je novotvar pro označení způsobu řešení matematických problémů pomocí kolektivního hraní her. Při kolektivní (byť neuvědomé) spolupráci se uplatňuje synergický efekt tzv. kolektivní inteligence, která je již používán v řešení různých problémů.

V poslední době se tento přístup využívá v oblasti počítačových her k řešení náročných úkolů jako je např. hra: klasifikace galaxií (Galaxy Zoo), Foldit, která výrazně pomohla s hledáním léku na HIV, EyeWire (mapování buněk myši rohovky) nebo Phylo (řešení genetických chorob).

Struktura práce bude následující:

1. Zhodnocení současného stavu na poli gamesourcingu, kolektivní inteligence a počítačových her.
2. Stanovení oblastí, kde se tento přístup využívá a kde je zatím nevyužit, příp. málo využíván.
3. Navrhnout vlastní řešení vybraného problému.
4. Vytvořte program a otestujte jej.
5. Zhodnoťte výsledky v závěru.

Seznam doporučené odborné literatury:

- [1] Roudenský Petr, Vytváříme PC hry, Computer press
- [2] Jan Jirkovský, GAME INDUSTRY, INFOSFERA
- [3] Jan Jirkovský, GAME INDUSTRY 2, INFOSFERA
- [4] Mark Watson, Practical Artificial Intelligence Programming With Java, November 11, 2008
- [5] Edwin Wise, Hands-on AI with Java : Smart Gaming, Robots, and More, McGraw-Hill/TAB Electronics, ISBN: 978-0071424967, 2004
- [6] Joseph P. Bigus, Constructing Intelligent Agents Using Java: Professional Developer's Guide, Wiley, ISBN 978-0471396017
- [7] Mat Buckland, Programming Game AI By Example (Wordware Game Developers Library), Jones & Bartlett Learning, ISBN 978-1556220784
- [8] Jason Gregory, Game Engine Architecture, A K Peters/CRC Press, ISBN 978-1568814131
- [9] David Pallmann, Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++, Microsoft press, ISBN 978-0735605657
- [10] Ian Millington, Artificial Intelligence for Games, CRC Press, ISBN 978-0123747310

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Ivan Zelinka, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 2017



.....
Bc. Jiří Arleth

Poděkování

Děkuji své přítelkyni Bc. Andree Pravdové za pomoc s ilustracemi a gramatickou kontrolou této práce.

Abstrakt

Tato diplomová práce zkoumá gamesourcing, tedy hrou řízený crowdsourcing, ze všech možných úhlů, jako jsou historie, motivace či paradigma a uvádí několik současných příkladů projektů z praxe. Praktická část práce spočívá v využití shromážděných teoretických vědomostí k vytvoření hry kolektivně řešící optimalizační úlohu obchodního cestujícího na neorientovaném úplném grafu. Hra má podobu bludiště a umožňuje hráčům fungovat na principu podobném mravenčímu algoritmu. Její výkon je následně s těmito algoritmy srovnán. Všechny hráčské údaje byly monitorovány a uloženy pro případná budoucí využití.

Klíčová slova

Gamesourcing, Hra s účelem, Hra s užitekem, Hrou řízený crowdsourcing, Optimalizace, Problém obchodního cestujícího, TSP, Labyrint, Bludiště, Optimalizace mravenčí kolonií, ACO, Human Ant System.

Abstract

This master's thesis researches gamesourcing, i.e. game-driven crowdsourcing, from various points of view such as history, motivation or paradigm and presents several examples of contemporary projects of this kind. Practical part of the thesis utilizes previously gathered theoretical knowledge for development of a swarm game solving optimization version of the Travelling Salesman Problem on symmetric complete graph. The game has form of a maze enabling players to function similarly to Ant Colony Optimization algorithms. Its performance is then compared to those algorithms. All gameplay actions of the players were monitored and stored for possible future purposes.

Keywords

Gamesourcing, Game With a Purpose, Game-driven Crowdsourcing, Optimization, Travelling Salesman Problem, TSP, Maze, Labyrinth, Ant Colony Optimization, ACO, Human Ant System

Obsah

Seznam symbolů a zkratek	8
Seznam obrázků	9
Seznam grafů	10
Seznam tabulek	11
Úvod	12
1 Gamesourcing	13
1.1 Historie	13
1.2 Motivace	16
1.3 Paradigma	19
1.3.1 Struktura	19
1.3.2 Zábavnost	22
1.3.3 Věrohodnost výstupů	23
1.3.4 Hodnocení úspěšnosti	23
1.4 Současný stav	24
1.4.1 Astro Drone	24
1.4.2 EteRNA	26
1.4.3 Foldit	28
1.4.4 Play to Cure: Genes In Space	29
1.4.5 Sea Hero Quest	30
2 Experiment	32
2.1 Požadavky	32
2.1.1 Problém	32
2.1.2 Funkční specifikace	33
2.2 Návrh	40
2.2.1 Platforma	40
2.2.2 Vzhled	42
2.3 Implementace	45
2.3.1 Databáze	45
2.3.2 Backend	48
2.3.3 Frontend	52

2.4	Testování	59
2.5	Nasazení a údržba.....	59
3	Výsledky	60
3.1	Mravenčí algoritmy	60
3.1.1	Ant System	60
3.1.2	Elitist Ant System.....	61
3.1.3	Ranked Ant System.....	61
3.1.4	Best-Worst Ant System.....	61
3.1.5	Min-Max Ant System.....	62
3.1.6	Ant Colony System	62
3.2	Labyrint (Human Ant System)	62
3.3	Porovnání	62
	Závěr	66
	Reference.....	67

Seznam symbolů a zkratek

ACO – angl. Ant Colony Optimization, optimalizace mravenčí kolonií

AI – angl. Artificial Intelligence, umělá inteligence

ALP – angl. Average Lifetime Play, průměrná doba hraní za život

AM – angl. Ant Moves, mravenčí pohyby

CSS – angl. Cascading Style Sheet, kaskádové styly

ČR – Česká republika

DrD – Dračí doupě

ESA – angl. The European Space Agency, Evropská kosmická agentura

FAQ – angl. Frequently Asked Questions, často kladené dotazy

GWAP – angl. Games With A Purpose, hry s užitekem

HTML – angl. HyperText Markup Language

IE – angl. Internet Explorer, webový prohlížeč společnosti Microsoft

IF – Immortal Fighters

JSON – angl. JavaScript Object Notation, JavaScriptový objektový zápis

MySQL – databázový dotazovací jazyk

PC – počítač

PK – primární klíč

PHP – angl. HyperText Preprocessor, programovací jazyk

RNA – ribonukleová kyselina

RP – angl. role-playing, hraní na hrdiny

RPG – angl. Role-playing Game, hra na hrdiny

SVG – angl. Scalable Vector Graphics

TSP – angl. Travelling Salesman Problem, problém obchodního cestujícího

TV – televize

UC – angl. use-case, případ užití

UML – angl. Unified Modeling Language

Seznam obrázků

Obrázek 1 – Hodinky Johna Harrisona	13
Obrázek 2 – Vývoj loga společnosti Toyota	14
Obrázek 3 – Budova opery v Sydney	14
Obrázek 4 – Počítači	15
Obrázek 5 – Schéma hry shody výstupu	20
Obrázek 6 – Příklad průběhu hry shody výstupu	20
Obrázek 7 – Schéma inverzní hry	21
Obrázek 8 – Schéma hry shody vstupu	21
Obrázek 9 – Parrot AR.drone	25
Obrázek 10 – Ukázka z hraní Astro Drone	25
Obrázek 11 – Obrázky pořízené během letu (nahore) a jejich 5x5 px útržky (dole)	26
Obrázek 12 – Ukázka z hraní EteRNA	27
Obrázek 13 – Ribozym, zdroj	27
Obrázek 14 – Ukázka z hraní Foldit	28
Obrázek 15 – Ukázka z hraní Play to Cure: Genes In Space	29
Obrázek 16 – Data z DNA čipu	30
Obrázek 17 – Data z předchozího obrázku převedená do hry	30
Obrázek 18 – Ukázka z hraní Sea Hero Quest	31
Obrázek 19 – Úplný graf	32
Obrázek 20 – Hamiltonovská kružnice	32
Obrázek 21 – Diagram případů užití Labyrintu	34
Obrázek 22 – Pohled na klíčovou část webu Immortal Fighters – jeden z rozehraných příběhů	42
Obrázek 23 – Prvotní designový koncept Labyrintu	43
Obrázek 24 – Model databáze Labyrintu	45
Obrázek 25 – Příklad distribuce monster dle vah hran v grafu	50
Obrázek 26 – Příklad pro výpočet aktuálního skóre	52
Obrázek 27 – Hlavní stránka Labyrintu	54
Obrázek 28 – Detail vyskakovacího okýnka místnosti	55
Obrázek 29 – Chat Labyrintu	56
Obrázek 30 – Žebříček Labyrintu	57
Obrázek 31 – Sledování tras v Labyrintu	58

Seznam grafů

Graf 1 – Volný čas ve všední den populace ČR.....	16
Graf 2 – Volný čas o víkendu populace ČR.....	16
Graf 3 – Počet uživatelů Internetu v letech 1993 až 2016.....	17
Graf 4 – Činnosti provozované ve volném čase, ČR a 18 evropských zemí v roce 2007.....	17
Graf 5 – Procentuální zastoupení žánrů v letech 1975-2012.....	18
Graf 6 – Srovnání ACO algoritmů z hlediska potřebného počtu pohybů mravenců.....	65

Seznam tabulek

Tabulka 1 – Výklad životaschopnosti nestvůr	50
Tabulka 2 – Výsledky ACO algoritmů na problému Burma14.....	63
Tabulka 3 – Výsledky ACO algoritmů na problému Ulysses16.....	64
Tabulka 4 – Výsledky ACO algoritmů na problému Gr17	64

Úvod

Gamesourcing je nový termín, který vznikl spojením dvojice anglických slov game a crowdsourcing. Game znamená hra, vysvětlení pojmu crowdsourcing je však poněkud obsáhlejší. Jedná se o metodu, kdy je práce (často velice obtížná pro počítač, ale relativně snadná pro člověka) rozdělena mezi množství lidí a ti ji v závislosti na povaze zadání buďto společnými silami, nebo každý za sebe vyplní. V prvním případě se na konečném výsledku budou podílet všichni zúčastnění, v tom druhém bude vybrána práce jen toho nejlepšího. Spojením se hrou, tedy gamesourcing, dostaneme prostředek umožňující provádět crowdsourcing na trochu jiné bázi. A sice, že hráči ani nemusí vědět, že svým konáním nějaký crowdsourcing vůbec konají. A ještě se k tomu baví.

Historii a vývojem crowdsourcingu napříč časem a postupným zrodem gamesourcingu se zabývá první kapitola této práce. Dále popisuje motivaci, výhody a nevýhody hrou řízeného crowdsourcingu, základní paradigmaty při tvorbě takových her a několik příkladů těch v současnosti nejúspěšnějších.

Další dvě kapitoly jsou již ryze praktické. Cílem této práce je vedle teoretického průzkumu oblasti také gamesourcing reálně vyzkoušet na netriviálním matematickém problému. Druhá kapitola tak popisuje celý proces vývoje hry, která toto umožňuje formou kolektivní spolupráce hráčů. Od sběru požadavků až po nasazení a údržbu.

Finální kapitola potom nabízí srovnání výkonnosti hry a konkurence v podobě současných algoritmů používaných na vybraný problém. S lehkou mírou nadsázky se to dá nazvat jako jakýsi souboj mezi člověkem a počítačem.

1 Gamesourcing

V poslední dekádě se začíná Internetem nezadržitelně šířit nový fenomén – crowdsourcing, neboli využití velké skupiny lidí k nějaké tvůrčí činnosti. Např. čerpání nápadů, námětů, obsahu nebo příspěvků, ať už finančních, či odborných. Použití přitom může být prakticky jakékoliv: vzdělávání (Wikipedie), výzkum, zdravotnictví, doprava (hlášení nehod), marketing a reklama, dárcovství, dobrovolnictví... Ale i v politické sféře již našla „moudrost davů“ své uplatnění. Po roce 2008, kdy Island v důsledku hospodářské krize zkrachoval, došlo na tvorbu nové ústavy a její návrh byl poskytnut online k široké diskuzi. Debatami a připomínkováním skrze sociální sítě tak mohli islandští občané ovlivňovat a korigovat finální podobu a znění konstituční listiny (1) (2).

Crowdsourcing se tak pomalu do jisté míry stává součástí života každého z nás, aniž bychom si to mnohdy uvědomovali. Tato kapitola se věnuje popisu jedné jeho speciální podoby, tzv. gamesourcingu, kdy prostředkem ke kolektivní spolupráci je hraní her.

1.1 Historie

Termín crowdsourcing byl poprvé použit v roce 2006 (3), což by mohlo vést (vzhledem k rozmachu Internetu) k mylné domněnce, že se jedná výhradně o online záležitost. Proces využití mas ke spolupráci na nějakém úkolu ale fungoval již od nepaměti, pravěkem počínaje (4). Šlo tehdy sice jen o přijímání rad a doporučení náčelníkem od svých podřízených, ale určitým způsobem se vlastně jednalo o jakési kolektivní podílení se na dosažení cíle (např. jak přežít zimu). Jistá forma crowdsourcingu tedy existovala již desítky tisíc let před vznikem vševědoucího média, takže se v zásadě nejedná o žádnou novou věc.

Příkladem více podobným současné formě crowdsourcingu budiž případ z roku 1714, kdy britská vláda řešila bezpečnost v mořeplavectví. „Problém zeměpisné délky“ činil plavby velmi obtížnými a ročně stál námořnictvo jejího veličenstva tisícovky životů. Šlo o to, že v té době dokázali námořníci bez problému (podle výšky Slunce) určit svou zeměpisnou šířku, tedy to, jak daleko jsou na sever nebo na jih od rovníku. Neexistovala však žádná spolehlivá metoda, pomocí níž by určili zeměpisnou délku, tedy jak daleko na východ nebo na západ dopluli. Britský parlament tudíž vypsál odměnu 20 000 liber pro toho, kdo přijde s řešením. V přepočtu na dnešní hodnotu by taková suma odpovídala několika milionům amerických dolarů (5). Náročnost takového úkolu spočívala především v potřebě přesného měření času. Kyvadlové hodiny nepřicházely kvůli houpání se ve vlnách v úvahu, hodiny s pérem a ozubenými kolečky zas byly nespolehlivé a nepřesné. Výhercem soutěže se stal John Harrison, syn tesaře. Přišel s námořním chronometrem, tj. přesnými vakuově uzavřenými kapesními hodinkami. Ty se po 47denní plavbě zmýlily pouze o 39,2 vteřiny (6) a vynesly tak svému vynálezci tučný honorář. Tento příklad krásně ukazuje jednu z hlavních myšlenek crowdsourcingu – inovace a kreativita může přijít odkudkoliv.



Obrázek 1 – Hodinky Johna Harrisona

V roce 1879 tomu již bylo mnoho let od vzniku nápadu vytvořit chronologický slovník anglického jazyka. Původním záměrem bylo sepsat 4dílný, 6400stránkový slovník, obsahující významy slov v jejich chronologickém pořadí od nejstaršího citátu v kontextu až po nejnovější pro dokumentování změny ve významu. Mapovat se mělo až do dob Sasů, tedy zhruba 1000 let zpět. Jednalo se bezesporu o velmi ambiciózní projekt, který mohl být jen stěží dotažen do cíle pouze jediným člověkem. „*Bylo by nezbytné najmout tým – navíc velmi veliký – složený ze stovek a stovek neplacených amatérů pracujících dobrovolně*” (7 str. 106) (vlastní překlad)¹. Proto byla rozeslána obsílka vyzývající dobrovolníky, aby si vybrali historickou etapu, z které by si přáli číst, a následně posílali výpisky slov užívaných v té dané literatuře. Roku 1928 byl projekt dokončen a jeho výsledkem byl Oxford English Dictionary. Podílelo se na něm neuvěřitelných 6 000 000 čtenářů (6). Tento slovník je i v současnosti nadále rozšiřován prostřednictvím online formulářů a se svými více než 600 000 hesly v kontextu 2 500 000 citátů je jedním z největších a nejvýznamnějších slovníků vůbec (8).

Přehoupneme se do minulého století, kdy pro výsledek crowdsourcingového projektu, jenž se konal roku 1936, nemusíme chodit vůbec daleko. Prakticky jej máme denně na očích. V té době vyhlásila společnost Toyota veřejnou soutěž o redesign svého loga. Obdrželi celkem 27 000 návrhů (9). Ten vítězný vydržel automobilce až dodnes a můžeme jej vidat na každém z jejich vozů.



Obrázek 2 – Vývoj loga společnosti Toyota

Crowdsourcing je již nějakou dobu využíván i v architektuře. Roku 1955 vypsali premiér australského státu New South Wales – Joseph Cahill – soutěž o 5 000 liber tomu, kdo navrhne operní budovu na pobřeží zálivu v Sydney. Zasláno bylo dohromady 233 návrhů z 32 různých zemí. Vítězný návrh dánského architekta Jørna Utzona předznamenal vznik jedné z nejkoničtějších a nejnovativnějších staveb současné doby. Tento typ architektonických soutěží je i nadále hojně využíván (10).

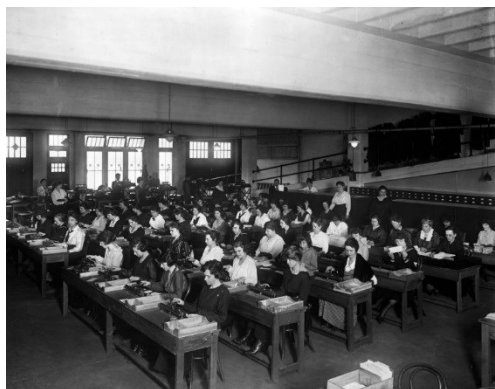


Obrázek 3 – Budova opery v Sydney

Do příchodu počítačů tak jak je známe dnes, byli za počítače označováni lidé, jejichž zaměstnáním bylo ručně počítat matematické výpočty. Hráli klíčovou roli především během 2. světové války (11). Každý

¹ “It would be necessary to recruit a team – moreover, a huge one – probably comprising hundreds and hundreds of unpaid amateurs, all of them working as volunteers.”

z nich počítal jednoduché úlohy v rámci týmu, který takto řešil již nějaký komplexnější problém. Jejich mravenčí práce přinesla pokroky ve vědě, průmyslu a zbraních, a nakonec vedla až k vynálezu strojových počítačů, kterými byli nahrazeni. Vše ale stále probíhalo offline.



Obrázek 4 – Počítači

První pokusy o online spolupráci započaly v 60. letech, kdy se stovky a někdy až tisíce vývojářů byly schopny podílet na vývoji open-source software (12). S pozdějším masivním rozmachem Internetu se crowdsourcingu otevřely zcela nové obzory. Spolupráce mas se najednou stala jednodušší, rychlejší a především přístupnější většímu kvantu lidí. S příchodem třetího milénia tak vznikly weby, které se sice za crowdsourcing oficiálně neoznačují, ale ve své podstatě jím jsou – např. YouTube nebo Wikipedia. Rovněž se začínají objevovat snahy o využití lidí pro tzv. machine learning – budování inteligentních systémů za pomoci lidského vnímání a inteligence, např. Open Mind Initiative (13). Dobrovolníci skrze Internet poskytovali odpovědi na otázky, které počítače nedokázaly vyřešit („Co je na tomto obrázku?“). Toto už se hodně blížilo gamesourcingu, ale byla zde dvě úskalí – spoléhání se na ochotu neplacených dobrovolníků investovat své úsilí a čas a neexistovala zde žádná záruka, že jejich příspěvky nebudou nesprávné.

Roku 2006 Jeff Howe ve svém článku poprvé použil termínu „Crowdsourcing“ (14) a od té doby se již jedná o často skloňované a veřejnosti čím dál známější slovo. Objevovat se začaly první čisté crowdsourcingové projekty jako třeba DesignCrowd (crowdsourcingem vytváření grafických návrhů, log, webů atd.) nebo Digg (crowdsourcingový agregátor zpráv a novinek). Rozvíjet se rovněž začal i crowdfunding – přispívání většího počtu jednotlivců menšími obnosy k cílové části na financování zajímavého projektu či produktu.

S rostoucím trendem herního průmyslu byl správně odhadnut jeho potenciál pro crowdsourcing a toho samého roku byla pod záštitou Googlu spuštěna zřejmě vůbec první gamesourcingová hra Google Image Labeler (předtím The ESP Game), jejímž autorem byl profesor z Carnegie Mellon University Luis von Ahn (12). Jednalo se o popisování obrázků zábavnou a chytlavou formou.

Počet podobných počinů od té chvíle pomalu narůstá a povědomí o gamesourcingu se začíná dostávat i mezi širší veřejnost. V loňském roce (2016) za to může především hra Sea Hero Quest, která měla reklamu i na tuzemských televizních stanicích. Jedná se o hru pro mobilní zařízení, pomocí které je pomáháno studiu demence. Od vydání neuběhl ještě ani rok, a už si autoři pochvalují výtečné výsledky.

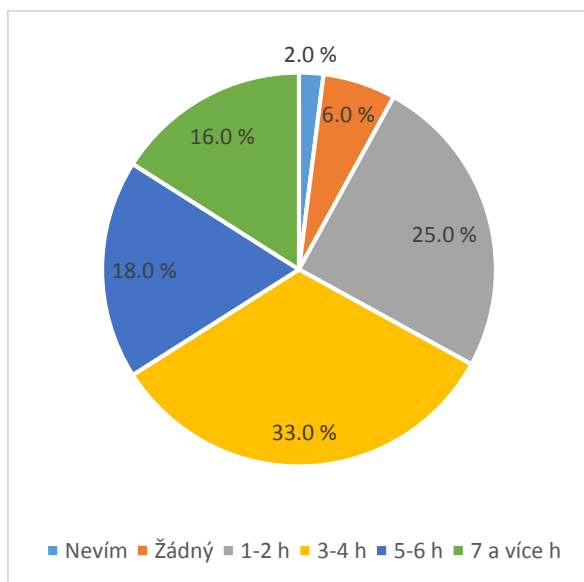
Zdá se, že gamesourcing je na vzestupu. Již v několika případech se prokázal jako výtečný nástroj (vizte 1.4). Jeho potenciál však ještě ani zdaleka nebyl plně využit...

1.2 Motivace

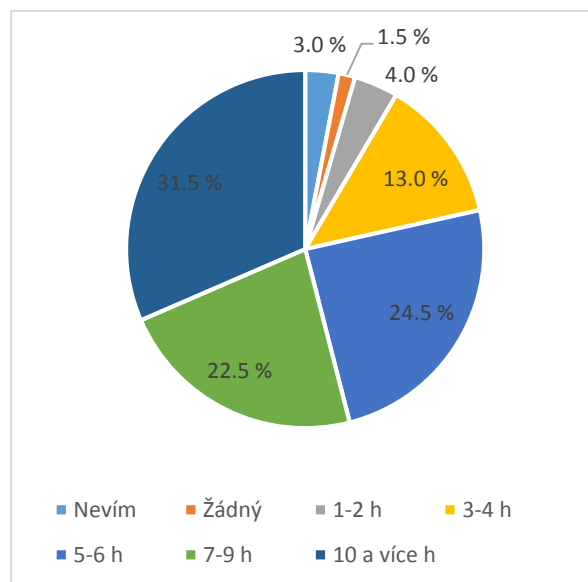
V čem je ale gamesourcing, tedy hrou řízený crowdsourcing, lepší než klasický crowdsourcing? Jaké jsou jeho přednosti, výhody a má naopak nějaké stinné stránky?

Volný čas

Princip crowdsourcingu tkví v dobrovolné účasti přispěvatelů. A to je hlavní kámen úrazu a nejvíce limitující faktor. Účastníci totiž musí obětovat svůj volný čas činnosti, ze které pro ně mnohdy ani neplyne žádný užitek. Což je v dnešní uspěchané době pro spoustu lidí nestravitelný problém. Dle výzkumu Sociologického ústavu akademie věd ČR v roce 2009 6 % české populace nemělo ve všední den volný čas žádný, čtvrtina pouze 1–2 hodiny, třetina 3–4 hodiny, 18 % 5–6 hodin a 16 % 7 a více hodin (15), což váženým průměrem vychází na cca 3,75 hodiny volného času na pracovní den. V sobotu nebo v neděli to u nás byl skoro dvojnásobek, zhruba 7 hodin. Při srovnání s evropskými odhady se jedná o přibližně stejná čísla (16). A to není mnoho.



Graf 1 – Volný čas ve všední den populace ČR, vlastní zpracování

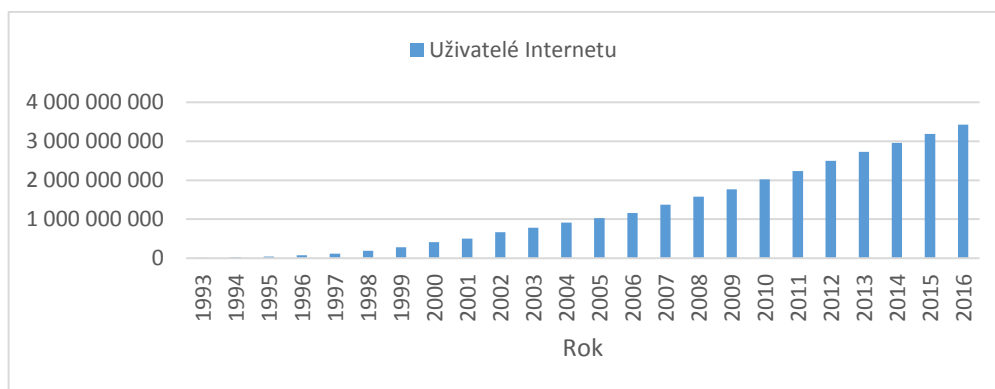


Graf 2 – Volný čas o víkendu populace ČR, vlastní zpracování

Internet

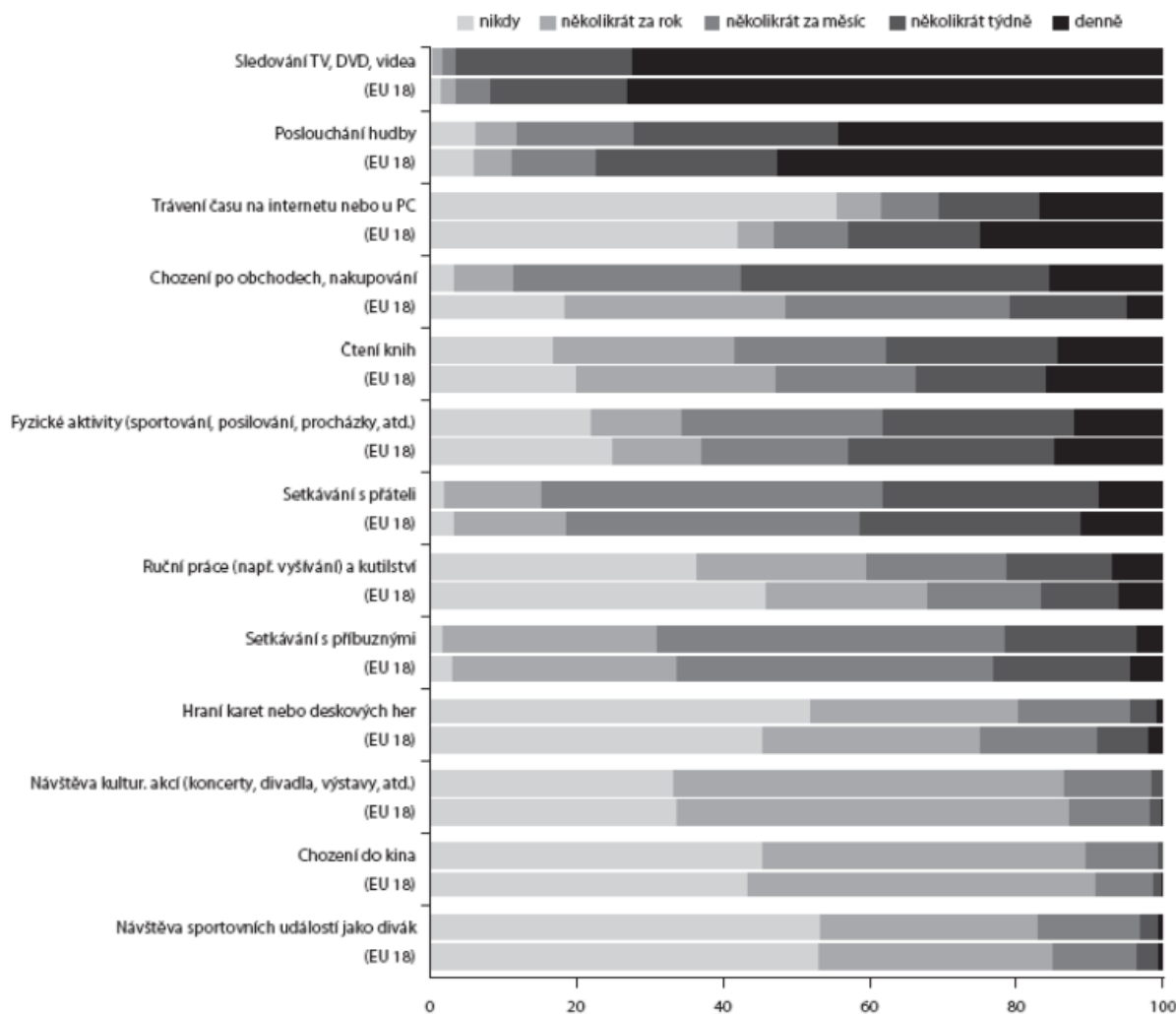
Když se však podíváme (vizte Graf 4), jak je daný čas (v Evropě i Česku) tráven, všimneme si, že po sledování TV a poslechu hudby jsou nejčastější činnosti Internet a PC (16). První řádek vždy popisuje údaj pro ČR, druhý pro evropské země². S přihlédnutím k faktu, že počet uživatelů Internetu neustále narůstá (vizte Graf 3) a s ohledem na priority v trávení volného času, můžeme tedy směle vyslovit tvrzení, že online svět je ideálním prostředím pro provoz crowdsourcingu.

² Rakousko, Belgie (Flandry), Bulharsko, Chorvatsko, Finsko, Francie, Německo, Maďarsko, Irsko, Litva, Norsko, Polsko, Rusko, Slovensko, Slovinsko, Švédsko, Švýcarsko, Velká Británie.



Graf 3 – Počet uživatelů Internetu v letech 1993 až 2016, vlastní zpracování

Současný počet internetových uživatelů již přesáhl hranici 3,5 miliard lidí, což činí téměř 50 % světové populace (17). Roku 1995 to bylo jen 1 %. Mezi lety 1999 a 2013 se počet uživatelů Internetu zdesetinásobil. Internetovým uživatelem se zde myslí osoba, která má k Internetu přístup skrze počítač nebo mobilní zařízení ze svého domova.



Graf 4 – Činnosti provozované ve volném čase, ČR a 18 evropských zemí v roce 2007, řádková procenta, zdroj: (15)

Hraní

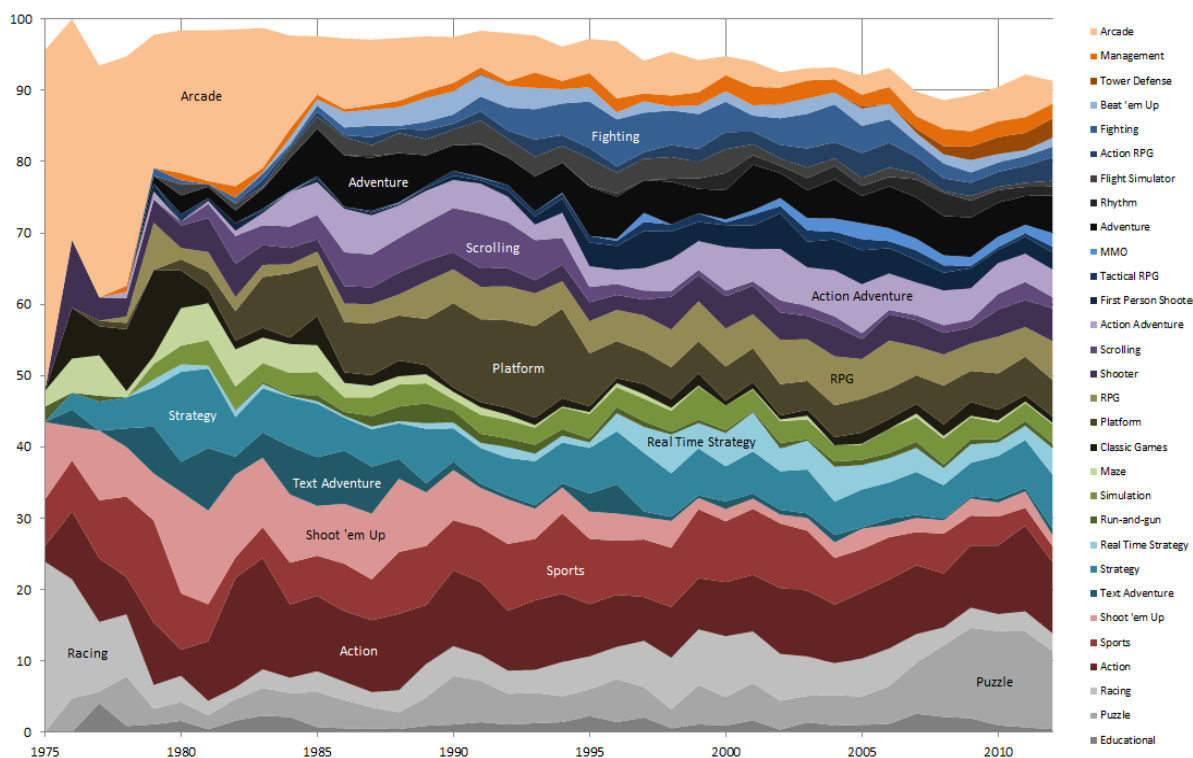
Dle zprávy herní společnosti Spil Games (18) se k roku 2013 na Zemi nacházelo 1,2 miliardy lidí hrajících hry. Z toho 700 milionů hrálo online. Právě díky stále větší dostupnosti Internetu se online hry začaly dostávat více do popředí. Za jejich úspěchem však především stojí kombinace jednoduchého ovládání, ale nesnadného ovládnutí, což představuje zajímavý mix zábavy a výzvy.

Hraní her se stalo mainstreamovou záležitostí, která již dávno není zatížena stereotypem osamělého teenagera zamčeného před světem v zatemnělém pokoji a sedícího před počítačem. Nejedná se nadále o činnost pouze pro mladé. Hraje kupříkladu 41 % žen a 37 % mužů nad 45 let. Což není moc velký rozdíl oproti kategorii 15 až 24 let, kde poměr hrajících činí 47 % žen a 54 % mužů (18).

K šíření her do starších generací nepochybně přispěl rozvoj mobilních technologií. 52,7 % současných uživatelů Internetu k němu přistupuje z mobilního zařízení (19). Takže hrát (online) se již dnes dá takřka kdekoli. V autobuse cestou do práce, při čekání na schůzku, na WC, ve volných chvílích na dovolené... Průměr takto stráveného času činí přibližně 40 minut denně (18).

A proč se vlastně tolik hraje? Jde především o zabavení se, ale to samotné by takový úspěch videoher nezaručilo. Hraje zde důležitou roli také snadně získaný pocit zadostiučinění při úspěchu, relax a rychlý únik od každodenního shonu a stresu. Motivace k opakovanému hraní se trochu liší na základě pohlaví. Zatímco mužům jde spíše o soutěživost, dosažení co nejlepšího umístění nad ostatními, ženám se jedná více o prozkoumání všech možností, dohrání celé hry (20).

Popularitu jednotlivých herních žánrů napříč lety přehledně zobrazuje následující graf (Graf 5) vytvořený na základě procentuálního počtu vydaných titulů daného žánru v daném roce.



Graf 5 – Procentuální zastoupení žánrů v letech 1975-2012, zdroj: Geek.com

Vidíme, že stále větší oblibě se těží puzzle hry, tzn. různé logické hříčky a hádanky, které jsou hlavně doménou mobilních her.

Čas a úsilí věnované hraní online her v sobě nese obrovský potenciál pro crowdsourcingová řešení problémů, na které je současná technika krátká. Spousta věcí pro člověka triviálních je totiž pro počítač takřka neřešitelná. Pokud by se gamesourcingem dokázalo využít těch 28 miliard minut (cca 500 mil. hodin) online hraní denně, mohlo by to vést k velkým věcem. A hráče by to přitom ani nic nestálo, dokonce by ani nemuseli vědět, že svým „oddychovým“ časem pomáhají něco vyřešit.

1.3 Paradigma

Žádná oficiální definice, jakou by gamesourcing měl mít strukturu a co vše obsahovat, neexistuje. Prof. Luis Von Ahn (s Laurou Dabbish) z Carnegie Mellon University, jenž je tvůrcem zřejmě první gamesourcingové hry The ESP Game, se však v jednom ze svých článků (12) zamýšlí, jak postup při tvorbě takovýchto her zobecnit. Označuje je zkratkou GWAP – Games With A Purpose, což lze volně přeložit jako hry s užitkem. Na základě svých mnohaletých zkušeností v této oblasti sestavil tři „šablony“, které vystihují trojici možných struktur her, jež lze podle něj aplikovat na jakýkoliv výpočetní problém. Každá šablona definuje pravidla hry a podmínky výhry tak, aby bylo v nejlepším zájmu hráčů provádět požadované výpočty. K tomu uvedl sadu principů zvyšujících zábavnost a atraktivitu her s užitkem a přitom zajišťujících kvalitu jejich výstupů. Nakonec představil metriky posuzující úspěšnost gamesourcingové hry z hlediska užitku získaného za člověkohodinu hraní.

1.3.1 Struktura

Motivace gamesourcingu je založena na třech hlavních faktorech (vizte 1.2):

- stále se zvyšující procento populace s přístupem k Internetu,
- některé úkony jsou neřešitelné počítači, ale snadné pro člověka,
- lidé tráví spoustu času hraním počítačových her.

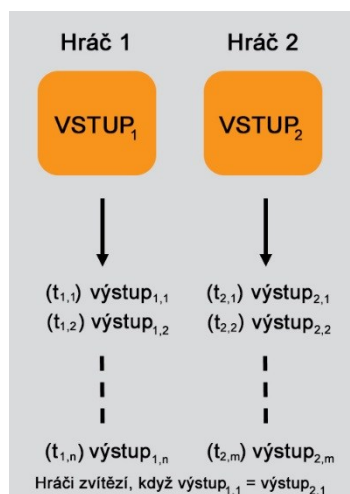
Gamesourcing na rozdíl od klasického crowdsourcingu nespočívá na altruismu či finanční motivaci, aby dotlačil lidi ke spolupráci. Sází na jejich chuť nechat se bavit. Gamesourcingová hra tedy musí být v prvé řadě zábavnou a teprve jako vedlejší efekt se při jejím hraní řeší nějaký užitečný výpočet. Herní mechanismus tak musí být úzce napojen na výpočetní problém způsobem vstup-výstup. Hráči je předložen vstup, a on na jeho základě vytvoří výstup, který potřebujeme.

Hra rovněž musí mít jasně určen cíl (podmínky vítězství) a pravidla vymezující, co hráči smí a nesmí dělat. Tyto prvky jsou velmi důležité, protože směřují hráče k provádění kroků, kterými budou řešit spjatý problém a zároveň by měly zajišťovat korektnost výstupu i v případě, kdy by se hráč snažil vyprodukovat výstup špatný.

Na základě těchto faktů lze definovat tři obecné typy struktur gamesourcingových her (12).

1. Shoda výstupu

Na začátku jsou k sobě náhodně vybráni hráči. V každém kole je každému z nich předložen stejný vstup a oni musí na jeho základě vytvářet výstupy. Hra tlačí hráče k tomu, aby se snažil o stejné výstupy jako jeho spoluhráči. Ti mezi sebou nemůžou nijak komunikovat a své výstupy navzájem nevidí.



Obrázek 5 – Schéma hry shody výstupu

Výhercem kola se stane ta dvojice, která jako první vyprodukuje stejný výstup. Nemusí k tomu dojít ve stejném čase, prostě jen musí jeden z hráčů vydat stejný výstup jako už některý spoluhráč před ním.

Na tomto principu byla vystavěna např. hra The ESP Game (12), kde byly vstupy obrázky a výstupy klíčová slova. S nemožností komunikace a vzájemnou anonymitou bylo tedy pro hráče nejlepší psát výstupy, které se skutečně obrázku týkaly. Hra přitom neříkala, že je nutné zadávat správná klíčová slova. Jen, že je třeba se pokusit o stejná, která zadá i spoluhráč. Jinými slovy: „Přemýšlej jako ostatní.“

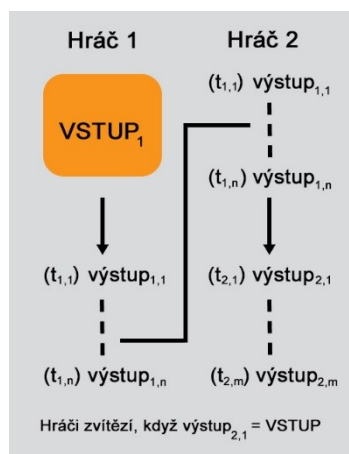


Obrázek 6 – Příklad průběhu hry shody výstupu

Tato herní struktura (shody výstupu) zároveň dobře zajišťuje korektnost výstupů, jelikož na výsledku se shodne dvojice zcela nezávislých zdrojů.

2. Inverze

Tímto způsobem jsou vystavěny např. hry Peekaboom, Phetch nebo Verbosity (12). Na začátku jsou k sobě náhodně přiřazeni hráči a jeden z nich je vybrán jako „popisovatel“, ostatní budou „hadači“. Popisovatel dostane vstup, na jehož základě vyprodukuje výstupy, které následně pošle hadačům. Ti se podle těchto poskytnutých informací snaží uhádnout vstup. Např. ve Verbosity je vstupem slovo a výstupy fakta k němu spjatá. Třeba u vstupu „mléko“ by výstupy mohly být „je to bílé“ nebo „souvisí to s kravami“.



Obrázek 7 – Schéma inverzní hry

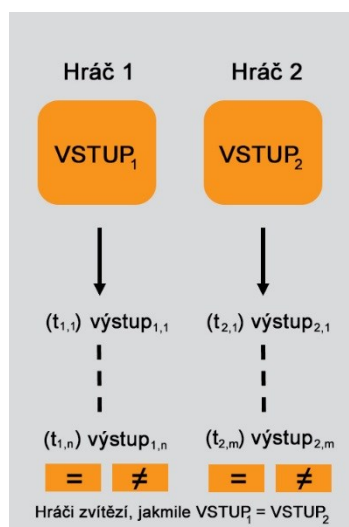
Vyhrává ten, kdo vstup trefí. Vítězem se ale rovněž stane popisovatel. Tato struktura hry jej tedy nutí snažit se vytvářet co nejpřesnější výstupy, aby měl co největší šanci, že na jejich základě někdo uhodne původní vstup. V podstatě se jedná o princip podobný klasické dětské hře na otázky.

Podoba hry může být i taková, že popisovatel hádané tipy vidí a může je hodnotit stylem „přihořívá/zima“. Taky je vhodné v případě odlišné zábavnosti rolí hadače a popisovatele zajistit, aby pravidelně docházelo k jejich střídání.

3. Shoda vstupu

Tato podoba je reprezentována např. hrou TagATune (12). Na začátku jsou k sobě náhodně sdruženi hráči a každému přidělen nějaký vstup. Hra ví, jestli jsou vstupy stejné, nebo ne, hráči však nikoliv. Následně musí hráči vytvářet výstupy, podle nichž v některou chvíli musí usoudit, zdali jsou jejich vstupy shodné, či ne. V TagATune jsou vstupem nahrávky/zvuky a výstupy tagy, pocity, popisy.

Vítězství dosáhne ten, který správně určí, jestli má s někým společný vstup, nebo naopak různý.



Obrázek 8 – Schéma hry shody vstupu

Aby se předešlo náhodnému tipování, měl by tento typ her silně penalizovat nesprávné odpovědi nebo třeba přidělovat bonusové skóre za správné odpovědi v řadě.

Všechny výše zmíněné struktury můžou mít i formu pro jednoho hráče. Lze to realizovat za pomoci AI oponenta, který by se choval na základě předdefinovaného scénáře. V případě her shody vstupu nebo výstupu je to poměrně jednoduché. Systém by při hře více hráčů mohl nahrávat jejich chování (výstupy a jejich časování) a toto následně replikovat při hře s jedním hráčem. U inverzních her je toto již obtížnější, protože je nutné dynamicky reagovat na konkrétní podněty popisovatele, popř. hadače.

1.3.2 Zábavnost

Asi nejdůležitějším aspektem (nejen) gamesourcingových her je zábavnost. Výstupy je třeba získávat takovým způsobem, aby se u toho hráči bavili. A když je to bude bavit, budou hrát dál a déle a produkovat více kýžených výstupů. Konkrétně zde bude vyjmenováno celkem pět základních principů, jak při gamesourcingu vyšší zábavnosti dosáhnout, ale obecně se dá říct, že zábavnost hry je přímo úměrná míře výzvy.

1. Časový limit

Nastavením časového omezení pro vytvoření výstupu lze zavést systém bonusů za rychlejší hráčské výkony apod. Z psychologie člověka je známo, že jasně vymezený a časově limitovaný úkol vede k vyšší výkonnosti než úkoly snadné a triviální (21). Hra však musí být dobře kalibrována a zadaný cíl musí v požadovaném časovém limitu opravdu představovat pro hráče výzvu. Taky je vhodné, aby zbývající čas byl viditelný.

2. Bodování

Asi nejprůchoďejší cesta k motivaci hráče vede skrze odměňování každého jím vytvořeného výstupu. Výsledné skóre po každé hře řekne hráči, nakolik byl úspěšný a vede ho k dalším pokusům, kdy se bude snažit svůj výsledek zlepšit.

3. Hodnosti

Systém hodností, kdy k získání vyšší hodnosti je nutné dosáhnout určitého počtu bodů/výstupů, je princip, který se na herní scéně osvědčil již dávno. Mnoho hráčů bude hrát jenom proto, aby postoupili na další titul/hodnost, a tak se odlišili od ostatních.

4. Žebříčky

Sestavování žebříčků nejúspěšnějších hráčů může být hráčům silnou motivací. Každý by chtěl alespoň na chvíli vidět své jméno na vůdčí pozici. Navíc lze jejich škálováním (hodinové, denní, měsíční) nabídnout různou obtížnost dosažení takového výsledku. Což u některých hráčů může značně protáhnout jejich dobu hraní.

5. Náhoda

Vstupy by měly být hráčům předkládány náhodně. Jejich obtížnost tím bude proměnlivá a hra si udrží své kouzlo jak pro veterány, tak i nováčky. Zároveň je tímto do hry vnesena nejistota v případě časového limitu, protože nikdy nebude předem jisté, jak rychle půjde úkol splnit.

Náhodný výběr spoluhráčů má také pozitivní vliv na zábavnost. Zajišťuje unikátnost každého hraní a podporuje chuť hrát znovu.

1.3.3 Věrohodnost výstupů

Základní výše uvedené struktury gamesourcingových her nemusí někdy zajistit dostatečně spolehlivé výsledky. Kdyby se hráči dopředu domluvili, že budou schválně všichni zadávat stejné a nesprávné výstupy (např. vždy „xxx“ při popisování každého z obrázků), což by v případě hry shody výstupu bylo bráno jako vítězný výsledek, systém by pak sbíral nesprávná data. Následující mechanismy by podobným typům tajně domluveného chování měly předejít.

1. Náhodné seskupování

Gamesourcingové hry jsou tvořeny pro stovky, ne-li tisíce hráčů hrajících zároveň a z různých koutů světa. Pokud budou k sobě hráči seskupováni náhodně, nevědí, s kým zrovna hrají, tudíž se nemůžou dopředu nijak dohodnout na podvádění.

2. Testování hráčů

Hráči může být občas předložen vstup, pro který jsou již všechny korektní výstupy známy. Pokud pro něj hráč vyprodukuje nesprávný výstup, stává se podezřelým a jeho dalším výstupům už se nemusí věřit. V závislosti na počtu takovýchto testovacích vstupů pak lze určit pravděpodobnost validního výstupu. Např. kdyby polovina ze všech vstupů prezentovaných hráči byly vstupy testovací (a na všechny by vydal korektní výstup), znamenalo by to, že ostatní jím vyprodukované výstupy jsou na 50 % důvěryhodné.

3. Opakování

Hra by měla být navržena způsobem, který nebude považovat výstup za korektní, dokud jej nedosáhne určitý počet hráčů. Je takto možné garantovat libovolnou pravděpodobnost věrohodnosti výstupu. Např. v případě hry shody výstupu akceptující výstup až po n dosažených shodách, přičemž každý výstup má 50% správnost (na základě předchozího testování hráčů) je celková pravděpodobnost důvěryhodnosti tohoto výstupu $1 - 0,5^n$.

4. Tabu výstupy

Pro problémy, kde vstup má široké spektrum možných výstupů, můžeme chtít dostatečně pokrýt celou množinu výstupů. Toho lze dosáhnout za pomoci tabu výstupů. Nejčtenější korektní výstupy hra zakáže a hráči je zobrazí. To jej donutí uvažovat jiným způsobem a dosáhnout tak jiného korektního výstupu, třeba nového nebo méně čteného.

1.3.4 Hodnocení úspěšnosti

Jak určit úspěšnost gamesourcingové hry? Pokud dvě hry řeší stejný problém, jak poznat, která z nich je lepší? Pokud bychom na ně pohlíželi jako na algoritmy, byla by rozhodujícím kritériem složitost, tedy počet kroků k dosažení výstupu. U hry ale nemusí být zřejmé, co lze považovat za jeden výpočetní krok. Je tedy zapotřebí poněkud odlišné perspektivy.

Níže jsou vypsány tři přístupy k hodnocení úspěšnosti gamesourcingové hry (12). Přestože nezahrnují některé aspekty pro hry důležité (jako třeba viralitu nebo „word of mouth“), jedná se o poměrně spolehlivé ukazatele jejich užitečnosti.

1. Propustnost

Základní propustnost je definována jako průměrný počet vyřešených instancí problému (dvojic vstup-výstup) za jednu člověkohodinu hraní. Princip většiny her spočívá v opakovaném plnění stejného úkolu, čímž se hráč postupně zlepšuje. Pokud bychom chtěli brát zřetel na učení a zlepšování (popř. zhoršování) hráče, je třeba propustnost počítat jako průměrný počet vyřešených instancí problému všemi hráči za určitý smysluplný časový úsek (12).

2. Průměrná doba hraní za život

Jelikož se jedná o hru, musíme vzít v potaz i nakolik je zábavná. Můžeme mít vysokou míru propustnosti, ale k čemu to, když hru nikdo nebude chtít hrát? Zábavnost ale nelze jednoduše změřit. Velice úzce se odvíjí od vzhledu a implementace hry. Stačí na první pohled nicotná změna do uživatelského rozhraní nebo bodovacího systému, a rázem může zábavnost vyšplhat nahoru. Metrika ALP (12), čili „Průměrná doba hraní za život“, je průměrem času stráveného hraním dané hry všemi hráči, kteří ji kdy hráli.

3. Očekávaný přínos

Jakmile známe průměrný počet instancí problému vyřešený za člověkohodinu hraní (propustnost) a očekávanou dobu strávenou hraním naší hry (ALP), můžeme určit přínos každého hráče. Veličina očekávaného přínosu říká, kolik můžeme průměrně očekávat vyřešených instancí problému jedním hráčem. Získáme ji jako výsledek součinu propustnosti a ALP (12).

1.4 Současný stav

Gamesourcing v současné době není ještě příliš využíván, z mého pohledu se většinou jedná hlavně o experimentální počiny. Naprostá většina takovýchto her spočívá v práci s obrázky, ať už se jedná o prosté určování jejich obsahu (např. určování buněk nakažených malárií v Malaria Training Game³), prepisování focených textů (např. deníky vojáků z 1. světové války v Operation War Diary⁴), nebo označování zajímavých míst (např. trajektorie částic vzniklých po kolizi v obřím urychlovači částic v Higgs Hunters⁵). V drtivém množství případů se však ani tak nejedná o hry, jako spíš jen o vizuálně hezky zpracované, přesto ale stále čistokrevné crowdsourcingy.

Následující výčet obsahuje hry, které z tohoto průměru výrazně vybočují především svým zpracováním, které není tak průhledné a přímočaré. Právě fakt, že hráč by vlastně ani nemusel poznat, že hraním bude přispívat k řešení problému, je jednou z nejzajímavějších vlastností, kterou gamesourcing může disponovat a která v obyčejném crowdsourcingu chybí.

Projekty jsou vyjmenovány v abecedním pořadí a u každé je popsán problém, který řeší, jakým způsobem jej hra pojala a výsledky, kterých zatím dosáhla.

1.4.1 Astro Drone

Tuto hru vytvořila Evropská kosmická agentura (ESA) za účelem výzkumu, zdali je možné určit vzdálenost objektu jen na základě pohledu na statický obraz. Základní premisou je předpoklad, že při

³ Dostupné na: <http://biogames.ee.ucla.edu/>

⁴ Dostupné na: <https://www.operationwardiary.org/>

⁵ Dostupné na: <https://www.higgshunters.org/>

přibližování se k objektu dochází k menšímu kolísání barev a textur. K měření těchto změn je využíváno pravděpodobnostní rozdělení předpokládaného výskytu rozdílných textur, na jehož základě je pak vypočítána Shannonova entropie⁶. Na sérii 385 obrázkových sekvencí se ukázalo, že v 88,6 % případů dochází k poklesu entropie při přibližování se k překážce. Použitá data však nebyla ideální, jelikož u většiny sekvencí se jednalo pouze o simulaci přibližování za pomoci škálování jednoho obrazu. Ta zbývající byla vytvořena z nahrávek ručních kamer, což ale vedlo k limitaci pohybu a lokací (22).



Obrázek 9 – Parrot AR.drone, zdroj: Parrot.com

Cílem projektu Astro Drone je získat větší a různorodější množství dat pro analýzu tohoto problému, který by mohl mít využití při navigaci a přistávání vesmírných sond a robotů.

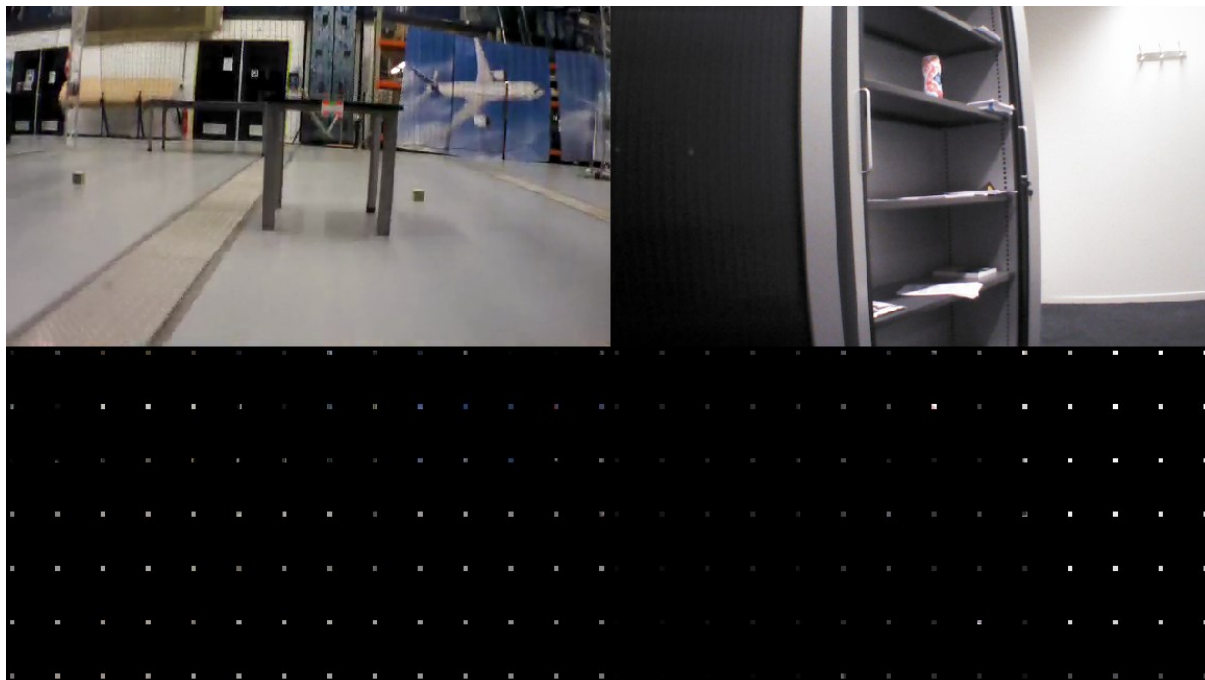


Obrázek 10 – Ukázka z hraní Astro Drone, zdroj: www.esa.int

K účasti je zapotřebí kvadrokoptéra Parrot AR.drone (u nás se cena pohybuje okolo 8 000 Kč), iPhone nebo iPad a mobilní aplikace Astro Drone, která je ke stažení zdarma. Samotná hra pak využívá kamery

⁶ Spočítá se jako $H = -\sum_i p_i \log_b p_i$, kde p_i je pravděpodobnost výskytu daného znaku a b počet znaků v abecedě.

dronu v rozšířené realitě a v současnosti obsahuje dvě možné úrovně. První simuluje dokování modulu do Mezinárodní vesmírné stanice a druhá let vesmírné sondy Rosetta skrze kosmický odpad ke kometě 67P/Churyumov–Gerasimenko. Ta následně musí ve správnou chvíli vypustit modul Philae tak, aby bezpečně na kometě přistál. Tento proces je poté ohodnocen ve formě skóre.



Obrázek 11 – Obrázky pořízené během letu (nahore) a jejich 5x5 px útržky (dole), zdroj: www.esa.int

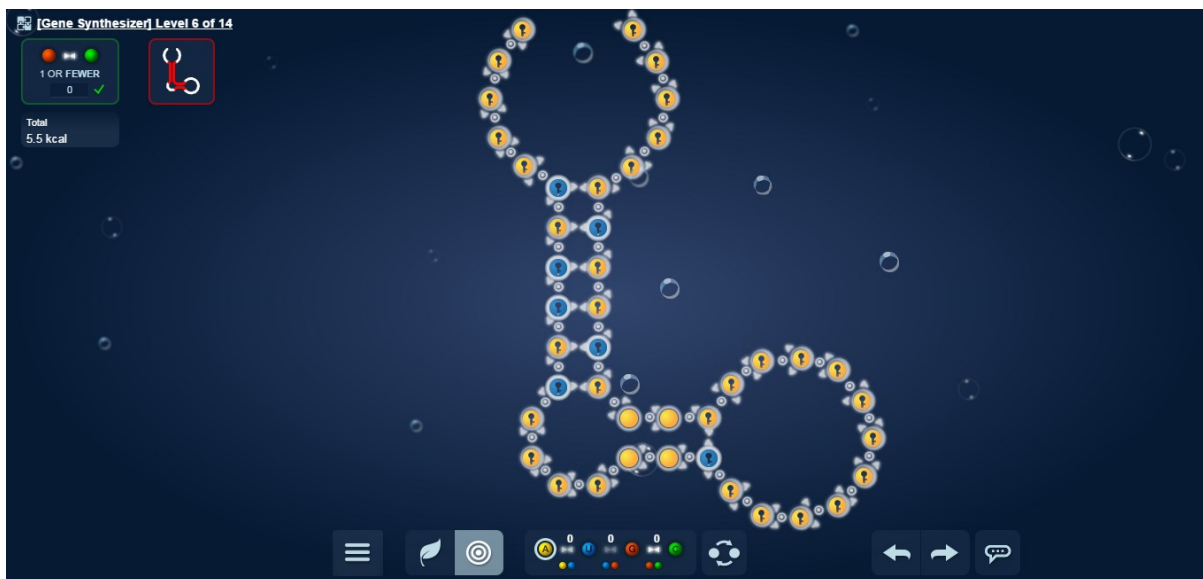
Během hraní je rovnoměrně (z hlediska vzdálenosti od „komety“ či „stanice“) pořízeno 10 snímků. Pokud se hráč rozhodne své skóre započítat, souhlasí tím zároveň s odesláním dat ESA. Snímky však z důvodů ochrany soukromí nebudou poslány celé, nýbrž jen jako síť 5x5 pixelových útržků. Aby agentura získala i nějaké informace o struktuře prostředí, ve kterém se hra odehrála, pošle se i jeden z obrázků převedený do černobílé, proceděný přes Hanningovo okno a transformovaný diskretní Fourierovou transformací. Tím je zajištěna nemožnost zpětného vyzískání původního obrazu a přitom zachování informací o jeho geometrické struktuře (22).

1.4.2 EteRNA

V roce 2010 spustily Carnegie Mellon University a Stanford University tento gamesourcingový projekt, v němž se jedná o skládání různých tvarů molekul RNA. Tvar RNA molekuly je klíčový v mnoha ohledech. Změnou tvaru můžeme přímo pozměnit chování molekuly, např. uzavřít problémový gen tak, aby se již nepřepisoval dále. Hraje také roli při interakci s enzymy nebo pokud se samy jako enzymy chovají. Pak se jedná o ribozymy. Jejich schopnost přeměňovat se v závislosti na chemických podmínkách je činí velmi užitečnými. Mohly by se „zabalit“ do tvaru efektivního pro putování naším tělem, aby se následně po vstupu do cílové buňky rozbalily. Cílem EteRNA je tedy ovládnout a vytvořit postupy k syntéze RNA do požadovaných tvarů (23).

Jedná se o prohlížečovou hru a probíhá tak, že hráči jsou předloženy báze a tvar, do kterého je třeba je uspořádat. Hráč musí pomocí myši tyto báze „označit“ jako A (Adenin), C (Cytosin), U (Uracil), nebo G (Guanosin), a ty se začnou dynamicky spojovat a postupně utvářet obrazec. Každé dvě báze mezi

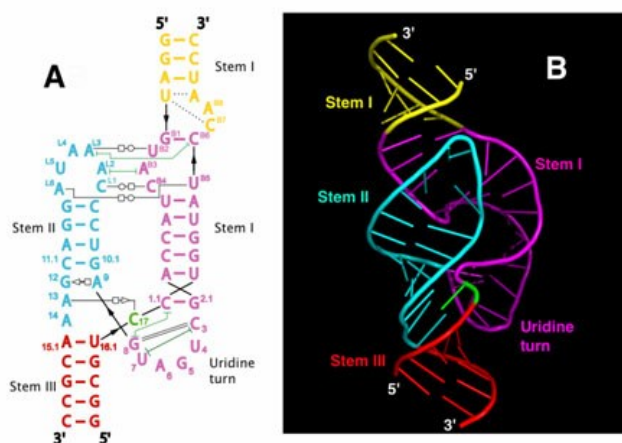
sebou vytváří různě silné vazby, a navíc je ve hře mnohdy přidán limitující faktor typu „použij maximálně jednu vazbu G-C“. Cílem je dosáhnout požadovaného tvaru s co nejnižší energetickou náročností, tzn. s co nejslabšími vazbami. Od toho se pak odvíjí finální skóre.



Obrázek 12 – Ukázka z hraní EteRNA

Když hráč vyřeší určitý počet úkolů, odemkne se mu možnost vytvářet vlastní návrhy tvarů molekul pro ostatní hráče. Zvláště zajímavé či slibné počiny mohou být autory vybrány k reálné syntéze v laboratoři.

Stejným problémem skládáním RNA se již zabývá program Folding@Home, který využívá výpočetní kapacity dobrovolníků s PlayStation 3, na kterých jsou potřebná data pasivně zpracovávána. EteRNA ale díky lidskému faktoru vykazuje podstatně lepší výsledky. Ne rychlejší (jednu instanci tohoto problému dokáže výkonný počítač zpracovat do minuty, člověku to zabere i dny), ale přesnější, což se prokázalo porovnáním při skutečném sestavování počítači a lidmi navrhnutých molekul v laboratorních podmínkách. Lidské návrhy byly v 99 % procentech lepší (23).



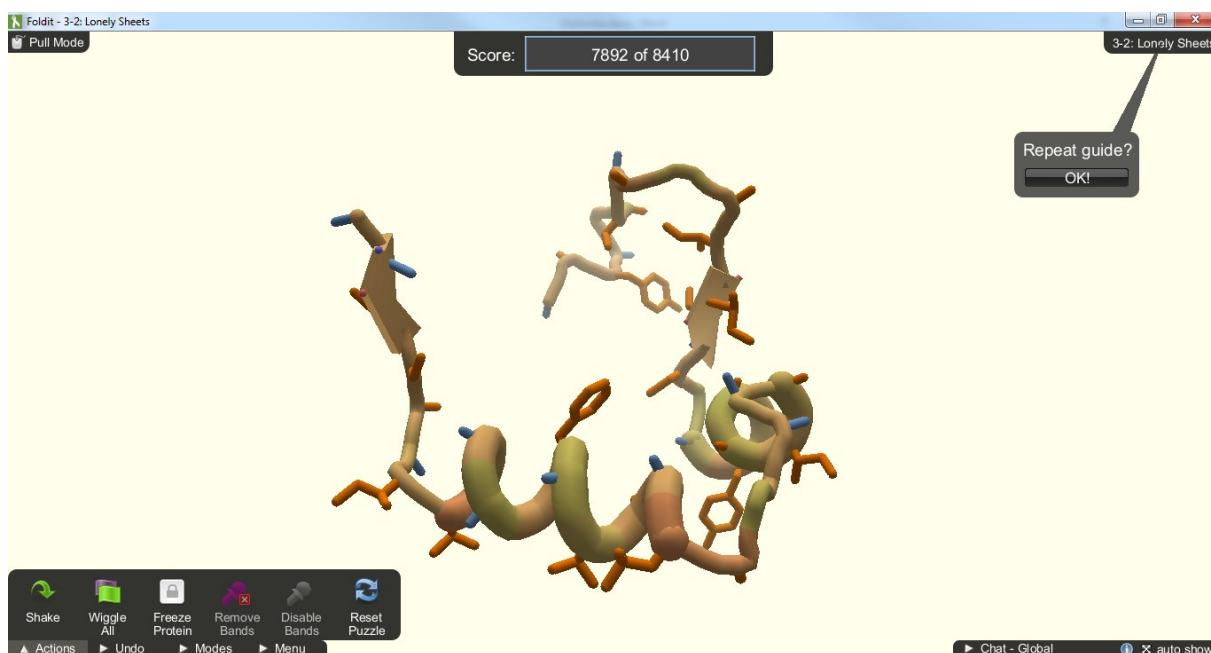
Obrázek 13 – Ribozym, zdroj: Geek.com

EteRNA disponuje více než 130 000 uživateli a tvůrcům se již na základě sesbíraných dat podařilo vyvinout několik nových algoritmů. První z nich – EteRNAbot, tehdy předčil všechny ostatní algoritmy v 19 z 20 případů (23).

1.4.3 Foldit

Proteiny jsou součástí téměř všech procesů v našem těle. Rozkládají cukry na energii, zpracovávají jídlo, posílají řídicí signály do mozku, přenášejí živiny krví. Mnoho z nich funguje jako enzymy, tzn. katalyzátory chemických reakcí (nejen těch prospěšných), které by jinak nemohly proběhnout. Existují tisíce různých proteinů, spousta je jich zodpovědná i za nemoci, ale všechny mají jedno společné – skládají se z dlouhého řetězce aminokyselin. Těch je jen 20 druhů a tvoří vždy dlouhou jednodlitou linii s malými větévkami. Protein vznikne pospojováním těchto linií do jedné hlavní (páteří). Klíčový je tvar, do jakého je tato linie zamotána. Protein se snaží tvořit co nejkompatnější objekt, takže ve výsledku jsou některé aminokyseliny uvnitř a některé vně a taky spolu různě sousedí. Tím je pak dána funkce každého proteinu (24).

V desktopové (Windows/OSX/Linux) hře Foldit, která vzešla z hlav vědců z University of Washington, se ohýbáním a tvarováním proteinu snažíte dosáhnout co nejvyššího skóre. Je třeba dbát o to, aby mezi jednotlivými tělesy nebyly moc malé, ale ani moc velké mezery, aby celkový protein byl co nejkompatnější, oranžové (hydrofobní) větévkami aby byly schovány co nejvíce dovnitř, modré (hydrofilní) zase venku, používání vodíkových můstků přidává více bodů atd. Vše se odehrává ve 3D prostředí a je ovládáno za pomoci myši, kterou můžete jednotlivé části proteinu ohýbat, posouvat a rotovat.



Obrázek 14 – Ukázka z hraní Foldit

Cílem tohoto projektu je nalézt prozatím neznámé struktury některých proteinů a popsat strategie, které přitom lidé využívali. Tím by se dalo dosáhnout významného pokroku při predikci proteinových struktur. Kupříkladu, během tří týdnů se podařilo určit strukturu proteinu, který je zodpovědný

za replikaci viru podobného AIDS v tělech opic. Tímto problémem se vědci předtím bezúspěšně zabývali 10 let (25).

Dále je zde také možnost pomoci k vytváření proteinů zcela nových, „na míru“ proti určité nemoci nebo za určitým účelem apod.

1.4.4 Play to Cure: Genes In Space

V této hře pro mobilní zařízení od Cancer Research UK se ocitnete v kokpitu vesmírné lodi, jejímž cílem je sbírat cennou komoditu zvanou „Element Alpha“. Za přivezený Element Alpha jste odměněni kredity, za které můžete svou loď různě vylepšovat, třeba posílit její motory nebo koupit lepší lasery na sestřelování asteroidů, které se vám při misích mohou připlést do cesty. Zároveň za své počínání získáváte zkušenosti, za které jste pak povyšováni do vyšších vojenských hodností.

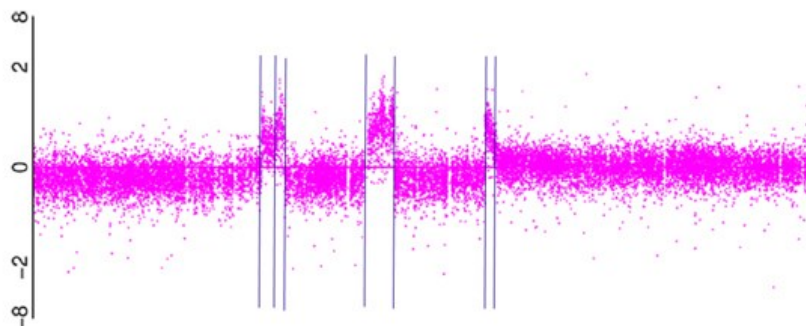


Obrázek 15 – Ukázka z hraní Play to Cure: Genes In Space

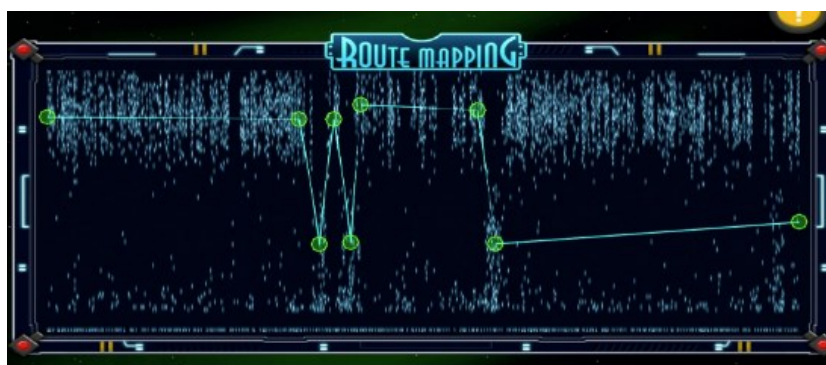
Zásadní moment celé hry se však odehrává těsně před začátkem každé mise. Je vám předložen sken oblasti znázorňující shluky Elementu Alpha a musíte si naplánovat trasu takovou, abyste proletěli co největšími nalezišti. Ve skutečnosti se však jedná o přemapovaná data z DNA čipu.

DNA se skládá z dvojic bází A, C, G a T uspořádaných do dvoušroubovice. Posloupnost bází je důležitá pro správné fungování našeho organismu. Pokud jsou z nějakého důvodu v DNA některé části složeny jinak, než by měly, dochází ke genetickým vadám. Problémem je, že těchto odlišností je spousta a mohou způsobit ztrátu nebo zisk různých genů, nejen toho, který nás zrovna zajímá a kterými jsou v tomto případě geny zodpovědné za rakovinu. Aby se určily příslušné segmenty, je nutné porovnávat vzorky DNA od spousty pacientů nakažených rakovinou. Využívá se k tomu DNA čipů, které dokáží srovnávat tisíce vzorků naráz. Tím ale vyvstává další problém – obrovské množství dat ke zpracování (26).

Existují softwary, které to dokáží, ale nic se nevyrovná vnímání lidského oka. Na obrázku níže (Obrázek 16) je výstup z DNA čipu. Růžový horizontální pás je částí zkoumaného DNA a jakékoliv výkyvy značí počty daných jeho segmentů navíc. Vidíme zde čtyři takto významné oblasti (jsou vyznačeny). Je ale nutné přesně určit, kde začínají a kde končí. A k tomu slouží ono plánování trasy rakety před startem mise. Hra tento výstup pěkně převede a už je jen na hráčích, aby dané oblasti vyznačili za účelem co největšího zisku Elementu Alpha.



Obrázek 16 – Data z DNA čipu, zdroj: scienceblog.cancerresearch.org



Obrázek 17 – Data z předchozího obrázku převedená do hry, zdroj: scienceblog.cancerresearch.org

Hraní této hry tedy značně urychluje analýzu příčin různých typů rakoviny a potažmo výzkum nových léků proti nim.

1.4.5 Sea Hero Quest

Demence výrazně zhoršuje kvalitu života a podle odborných odhadů postihne do roku 2050 135 milionů lidí. Sama o sobě není nemocí. Demence je termín používaný k souhrnnému označení příznaků, které se objeví v okamžiku, kdy mozkové buňky přestávají správně fungovat. Poškození těchto buněk narušuje schopnost mozkových buněk komunikovat mezi sebou navzájem. Pokud spolu mozkové buňky nemohou normálně komunikovat, má to dopad na myšlení, chování a pocity postiženého člověka (27).

Ačkoliv demence postihuje hlavně starší lidi, nelze ji považovat za nedílnou součást stárnutí. Demence si nevybírá – jde o chorobu, která může postihnout kohokoliv bez ohledu na jeho původ, vzdělání, životní styl nebo zdravotní stav. Neexistuje žádná léčba, která by onemocnění způsobující demenci dokázala zastavit. Zatímco některé procedury mohou trpícím alespoň trochu zkvalitnit život, léčba, která by zpomalila nebo zcela zastavila onemocnění, jako je i Alzheimerova choroba, neexistuje. To znamená, že se v průběhu času onemocnění zhoršuje (27). Jedinou nadějí tak zůstává objev nového způsobu léčby.

Hráč Sea Hero Questu se stává mořeplavcem, který brázdí vody moří a oceánů a fotí jejich obyvatele. Při tom uplatní zejména svůj postřeh a smysl pro orientaci. Právě zhoršení schopnosti prostorové orientace je jedním z průvodních jevů Alzheimerovy choroby, která je v 60–70 % příčinou demence. Hraním hry lidé vědcům pomohou tomuto procesu lépe porozumět a rozlišit, kdy je zhoršení schopnosti orientace přirozeným důsledkem stárnutí a kdy už symptomem nemoci. Na základě toho pak bude možné vyvinout nové a mnohem přesnější diagnostické testy na demenci a pokročit v její léčbě (27).



Obrázek 18 – Ukázka z hraní Sea Hero Quest

Jedná se o hru pro mobilní zařízení a dokonce je i v české lokalizaci. K dispozici je spousta postupně se odemykajících úrovní se stupňující se obtížností. Před vyplutím je hráči předložena mapa oblasti a umístění bójek, kterými se musí proplout v zadaném pořadí. Poté mapa zmizí a hráč je zcela odkázán na svou paměť. V závislosti na rychlosti splnění tohoto úkolu je pak odměněn hvězdami, za které lze svou loďku všelijak zkrášlovat. V jiné variantě jste zase dovedeni na cílové místo a musíte z něj vzdušnou čarou určit směr vašeho startovního bodu.

Po dokončení každého kola jsou data odeslána na University College London k analýze. Pouhé 2 minuty hraní poskytnou stejné množství dat jako 5 hodin v případě obdobného laboratorního pokusu. Hry se již zúčastnilo více než 2,5 milionu hráčů a nahráli dohromady 63 let herního času.

2 Experiment

Zde je popsán proces vzniku, tvorby a vývoje gamesourcingového experimentu a následná analýza jeho výsledků. Jedná se o webovou hru zakomponovanou do prostředí již existujícího herního portálu. Problém obchodního cestujícího je převeden do podoby bludiště a hráči jeho kolektivním procházením napodobují chování mravenčího algoritmu – evoluční techniky, která je v současnosti na tento problém hojně a s úspěchem aplikována. Cílem je sbírat data o pohybech hráčů, z nichž pak bude možné vyčíst jejich strategie a taktiky, které by mohly vést k vylepšení mravenčích algoritmů.

2.1 Požadavky

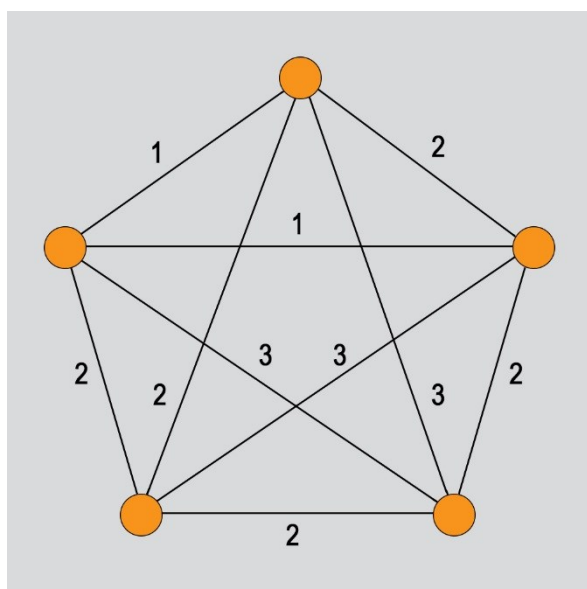
Pro gamesourcing je klíčové umět dobře spojit problém, který chceme řešit, s hrou. Musí mezi nimi existovat velice úzké vazby. Je třeba, aby hra svým provedením vedla hráče přesně k tomu, čeho chceme dosáhnout, tzn. k řešení problému, a to tak, aby toho byli schopni i laikové problematiky neznalí.

2.1.1 Problém

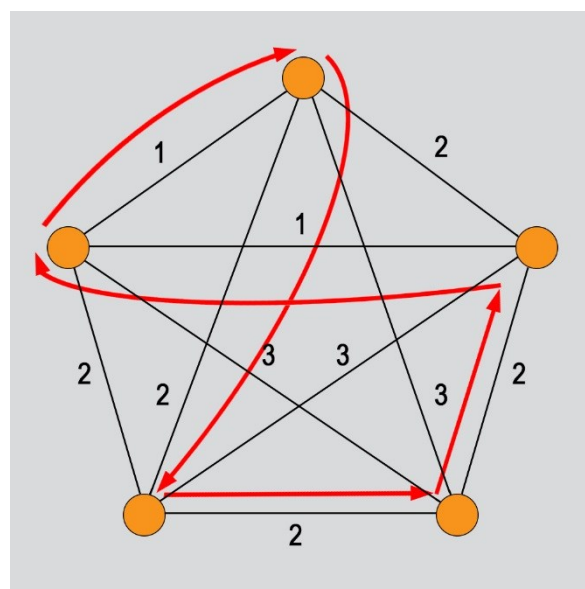
V tomto případě jsem se rozhodl pro řešení problému obchodního cestujícího a inspirovat se mravenčími algoritmy (vizte níže). Jednotliví mravenci však jsou zastoupeni lidmi. To by mohlo přispět k nalezení nových variant a úprav těchto algoritmů, a vést tak k jejich zefektivnění.

Problém obchodního cestujícího (TSP)

Název této úlohy vychází z jeho vzniku – obchodník potřeboval učinit okružní jízdu určitými městy (každé navštívit jen jednou) a vrátit se zpět domů. Přitom chtěl být ale co nejefektivnější, což znamená, že si cestu naplánoval tak, aby byla co nejkratší. V současnosti se TSP objevuje např. při vrtání u tištěných spojů, vyzvedávání zásilek na skladě nebo propojování komponent na základové desce počítače (28). Jedná se o NP-těžký problém (29). Tzn., že všechny problémy z množiny NP jsou na něj převeditelné v polynomiálním čase, přičemž on sám v NP není.



Obrázek 19 – Úplný graf



Obrázek 20 – Hamiltonovská kružnice

Vyjádříme-li tuto úlohu grafem, pak jeho vrcholy (uzly) reprezentují výchozí místo a místa, která má obchodník cestující navštívit. Každé dva vrcholy jsou v grafu spojeny hranou, která je ohodnocena vzdáleností, jíž je zapotřebí urazit, aby se obchodník dostal z místa reprezentovaném prvním uzlem do místa, jenž reprezentuje uzel druhý. Jedná se tedy o neorientovaný kompletní (úplný) graf. Cílem je potom nalezení nejkratší hamiltonovské kružnice (cyklu).

Všech možných hamiltonovských kružnic v grafu (na výchozím bodě nezáleží) je $\frac{(n-1)!}{2}$, takže počet možností nutných k prozkoumání velice rychle narůstá. Existuje několik exaktních algoritmů (Brute-force, Branch & Bound, Cutting Plane), které dokáží TSP vyřešit, ale v rozumném čase pouze u případů se 40–80 městy (30). Z tohoto důvodu našly hojně využití heuristické algoritmy, které negarantují správnost výsledku, ale ve většině případů je jimi nalezené řešení dostatečně kvalitní a hlavně je ho dosaženo v akceptovatelném čase. Jedním z těchto algoritmů jsou i mravenčí algoritmy neboli taky Optimalizace mravenčí kolonií (Ant Colony Optimization).

Mravenčí algoritmus (ACO)

Optimalizace za pomoci mravenčích kolonií je skupina algoritmů inspirovaná skutečným chováním mravenců v přírodě. Každý mravenec při své pouti vypouští feromony, které potom lákají ostatní. Čím silnější feromonová stopa je, tím více mravenců ji následuje. Hojně používané cestičky si tak neustále udržují vysokou hladinu feromonů, zatímco z těch nepoužívaných se nakonec feromony odpaří a mravenci už se jimi nevydají.

Aplikace na problém obchodního cestujícího vypadá následovně:

1. Každému mravenci je náhodně přiřazeno město.
2. Na základě rozhodovacího pravidla si mravenec vybere další město, kam se vydá.
3. Jakmile mravenec prošel všemi městy, vrátí se do toho počátečního.
4. Každý mravenec na svou cestu uloží množství feromonů, které se odvíjí od délky cesty. Kratší vyústí ve více feromonů.
5. Feromony se částečně odpaří.

Takto obecně probíhá jedna iterace ACO algoritmu. Různé verze mravenčích algoritmů pak mají různé způsoby tvorby rozhodovacích pravidel a odlišná pravidla pro feromony. Některé varianty mravenčích algoritmů jsou detailněji rozebrány v kapitole 0.

2.1.2 Funkční specifikace

Bylo zapotřebí vytvořit hru, která by šla dobře napasovat na řešení problému obchodního cestujícího za pomoci mravenčího algoritmu. Nejpřirozenější adaptací se mi jevilo bludiště. Uzly grafu reprezentovány místnostmi a hrany chodbami. Pokaždé se přitom bude jednat o úplný graf, tzn., všechny uzly budou navzájem propojeny hranami. Hráči budou vždy procházet bludiště společně, a to tak dlouho, dokud se některému z nich nepodaří nalézt nejkratší hamiltonovský cyklus. Je důležité, aby měli hráči přehled o pohybech ostatních, a to i těch předešlých. Na začátku budou váhy jednotlivých hran skryty, bude nutné jimi nejprve projít. Poté se již váha příslušné hrany zviditelní všem. Z důvodu navýšení motivace jsem se rozhodl ke hře zakomponovat žebříčky. Tím pádem vyvstala potřeba přijít s metrikou pro hodnocení kvality každé cesty. Pro vzájemnou komunikaci, ale především z důvodu kontaktu s adminy, bude k dispozici chat. Nejedná se však o klíčovou funkcionalitu, jde spíše jen o doplněk.

Role

Ve hře jsou přítomny tyto čtyři role:

- **Návštěvník:** Jedná se o uživatele, který se do hry zatím nezapojil, resp. nevstoupil do bludiště, nebo již hru hrál, ale momentálně je z bludiště venku (může se však kdykoli znova zapojit). Má možnost sledovat aktuální průběh hry (včetně současné pozice hráčů i minulosti jejich pohybů), stav žebříčků, smí přispívat do diskuze na chatu a zobrazit si pravidla hry.
- **Hráč:** Tohoto statutu nabyde návštěvník, jakmile vstoupí do bludiště. Od té chvíle může bludištěm procházet, dokud neuzavře hamiltonovský cyklus. Pak bludiště opustí a stává se z něj zpátky návštěvník. Hráč má všechna práva jako návštěvník.
- **Admin:** Má stejná práva jako hráč (a potažmo návštěvník), ale navíc může bludiště uzavírat (třeba z důvodů údržby nebo opravy systému), popř. otevírat. Také má na starost vytvoření nového bludiště, pokud předchozí bylo vyřešeno, nebo i z jakéhokoliv jiného důvodu.
- **Systém:** Zaznamenává informace o pohybech hráčů, dle vlastní metriky určuje jejich skóre a dává hráčům vědět o jeho změně. Aktualizuje žebříčky a vytváří nové dle potřeby. Při vyřešení bludiště některým z hráčů automaticky bludiště uzavře. Každý jednotlivý pohyb v bludišti zaznamenává do logu pro pozdější zpracování a analýzu.

Model případů užití

UC1 – Zobrazit chat

Možnost zobrazení stránky chatu, kde lze následně provádět další spjaté akce.

Aktéři: Uživatel, systém.

Základní tok:

1. Uživatel dá pokyn k zobrazení chatu.
2. Systém zobrazí stránku ve stylu diskuzního fóra. Tzn. textové pole a pod ním vložené komentáře.
3. *Rozšíření: UC2 Napsat komentář nebo UC3 Editovat komentář nebo UC4 Smazat komentář.*

UC2 – Napsat komentář

Systém umožní uživateli vložit komentář do chatu.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má zobrazen chat.

Základní tok:

1. Uživatel napíše příspěvek a dá pokyn k jeho odeslání.
2. Systém příspěvek ošetří. Odstraní případné HTML tagy, skripty a nahradí speciální znaky (např. pro změnu tloušťky nebo barvy písma, smajlíky) za příslušné prvky.
3. Systém zprávu uloží do databáze.
4. Systém nový komentář zobrazí.

Podmínky pro dokončení: Nový komentář bude korektně uložen v databázi.

UC3 – Editovat komentář

Systém umožní uživateli editovat svůj komentář v chatu.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má zobrazen chat a jedná se o jeho příspěvek.

Základní tok:

1. Uživatel dá pokyn k editaci svého komentáře.
2. Uživatel napíše úpravu svého příspěvku a dá pokyn k jeho odeslání.
3. Systém příspěvek ošetří. Odstraní případné HTML tagy, skripty a nahradí speciální znaky (např. pro změnu tloušťky nebo barvy písma, smajlíky) za příslušné prvky.
4. Systém aktualizuje zprávu v databázi.
5. Systém uloží původní zprávu do databáze.
6. Systém nový komentář zobrazí.

Podmínky pro dokončení: Komentář bude v databázi aktualizován a jeho původní verze bude v databázi rovněž zachována.

UC4 – Smazat komentář

Systém umožní uživateli smazat komentář v chatu.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má roli Admin.

Základní tok:

1. Uživatel dá pokyn k smazání komentáře.
2. Systém smaže daný příspěvek z databáze.
3. Systém smaže i všechny případné předchozí verze tohoto příspěvku.
4. Systém daný komentář již nebude zobrazovat.

Podmínky pro dokončení: Komentář bude z databáze odstraněn včetně všech jeho předchozích verzí.

UC5 – Zobrazit bludiště

Systém uživateli zobrazí bludiště.

Aktéři: Uživatel, systém.

Základní tok:

1. Uživatel dá pokyn k zobrazení bludiště.
2. Systém zobrazí bludiště, tzn. chodby a místnosti.
3. *Rozšíření: UC6 Zobrazit hráče nebo UC10 Vstoupit do bludiště.*

UC6 – Zobrazit hráče

Systém umožní uživateli zobrazit hráče nacházejícího se v místnosti bludiště.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má zobrazeno bludiště.

Základní tok:

1. Uživatel vybere místnost bludiště.
2. Systém zobrazí všechny hráče aktuálně se nacházející v této místnosti.
3. *Rozšíření: UC7 Zobrazit trasu.*

UC7 – Zobrazit trasu

Systém uživateli zobrazí trasu vybraného hráče.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má zobrazeno bludiště nebo žebříček a vybraného hráče. Ten již musel v bludišti vykonat alespoň jeden pohyb.

Základní tok:

1. Uživatel dá pokyn k zobrazení trasy zvoleného hráče.
2. Systém zobrazí stránku, kde trasu hráče zobrazí.

Alternativní tok:

- 2.1 Uživatel následně může z této stránky sledovat trasy i jiných hráčů.

UC8 – Zobrazit žebříčky

Systém umožní uživateli zobrazit žebříčky.

Aktéři: Uživatel, systém.

Základní tok:

1. Uživatel dá pokyn k zobrazení žebříčků.
2. Systém zobrazí žebříček pro aktuální bludiště obsahující informace o pořadí a odměnách.
3. *Rozšíření: UC7 Zobrazit trasu.*

Alternativní tok:

- 2.1 Uživatel si na této stránce může zobrazit i žebříčky předchozích bludišť.

UC9 – Zobrazit pravidla

Systém uživateli zobrazí pravidla hry.

Aktéři: Uživatel, systém.

Základní tok:

1. Uživatel dá pokyn k zobrazení pravidel.
2. Systém zobrazí stránku s pravidly.

UC10 – Vstoupit do bludiště

Systém uživateli umožní vstoupit do bludiště.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel v dané chvíli není uvnitř bludiště, má zobrazeno bludiště.

Základní tok:

1. Uživatel vybere místnost bludiště.
2. Uživatel dá pokyn ke vstupu do bludiště.
3. Systém zapíše uživatele do příslušné místnosti v databázi.
4. *Rozšíření: UC11 Pohyb bludištěm.*

Podmínky pro dokončení: Uživatel je v databázi zapsán do příslušné místnosti bludiště.

UC11 – Pohyb bludištěm

Systém uživateli umožní pohyb po bludišti.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel je uvnitř bludiště a má zobrazeno bludiště.

Základní tok:

1. Uživatel vybere místnost.
2. Systém vykoná UC12 Zobrazit váhu.
3. Uživatel vydá pokyn k přesunu do této místnosti.
4. Systém přesune hráče do dané místnosti.
5. Systém vykoná UC14 Ohodnotit dosavadní postup.
6. Systém vykoná UC13 Zalogovat info o pohybu.
7. *Rozšíření: Pokud uživatel dokončí hamiltonovský cyklus a nachází se ve výchozí místnosti, může vykonat UC16 Opustit bludiště.*

Alternativní tok:

- 3.1 Uživatel se rozhodne pro jinou místnost. Skok na krok 1.

Podmínky pro dokončení: Uživatel je přesunut do jiné místnosti a tento fakt je zanesen v databázi spolu s informačním logem. Uživatel je informován o svém bodovém zisku či ztrátě.

UC12 – Zobrazit váhu

Systém uživateli umožňuje zobrazit váhu zvolené hrany.

Aktéři: Systém.

Podmínky pro spuštění: Systém obdrží informaci, o kterou hranu má hráč zájem.

Základní tok:

1. Systém vyhledá hranu v databázi a zjistí její váhu.
2. Systém zobrazí danou váhu.

Alternativní tok:

- 2.1 Systém váhu nezobrazí, pokud tato hrana ještě nebyla nikým použita.

UC13 – Zalogovat info o pohybu

Systém umožňuje logovat potřebné informace o prováděných pohybech.

Aktéři: Systém.

Podmínky pro spuštění: Systém obdrží informaci o vykonaném pohybu hráče.

Základní tok:

1. Systém zaznamená čas, hráče, číslo pokusu, výchozí uzel, cílový uzel a výsledné celkové skóre hráče.

Podmínky pro dokončení: Log je uložen v databázi.

UC14 – Ohodnotit dosavadní postup

Systém umožňuje získat aktuální skóre hráče.

Aktéři: Systém.

Podmínky pro spuštění: Systém obdrží informaci o požadovaném hráči a čísle jeho pokusu.

Základní tok:

1. Systém vypočítá skóre.
2. *Rozšíření: Pokud uživatel získá maximální skóre, tak systém vykoná UC15 Nastavit přístup k bludišti a bludiště uzavře.*

UC15 – Nastavit přístup k bludišti

Systém umožňuje pro všechny hráče nastavit bludiště jako otevřené, nebo zavřené.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel je Admin.

Základní tok:

1. Uživatel (nebo systém) přepne aktuální stav přístupu bludiště. Z otevřeného na zavřený, nebo ze zavřeného na otevřený.

Podmínky pro dokončení: Fakt o změně přístupu k bludišti je aktualizován v databázi.

UC16 – Opustit bludiště

Systém uživateli umožní opustit bludiště.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel je uvnitř bludiště. Uživatel dokončil hamiltonovský cyklus a nachází se ve výchozím uzlu. Uživatel má bludiště zobrazeno.

Základní tok:

1. Uživatel vydá pokyn k opuštění bludiště.
2. Systém odstraní hráče z bludiště.

Podmínky pro dokončení: Hráč byl v databázi z bludiště odstraněn.

UC17 – Vytvořit bludiště

Systém umožňuje uživateli vytvořit bludiště.

Aktéři: Uživatel, systém.

Podmínky pro spuštění: Uživatel má roli Admin.

Základní tok:

1. Uživatel zadá počet uzlů a poschodí.
2. Uživatel vydá pokyn k vytvoření nového bludiště.
3. Systém se dotáže uživatele, zdali si je jist svým konáním.
4. Uživatel potvrdí své rozhodnutí.
5. Systém zkontroluje validitu zadaných hodnot.
6. Systém vytvoří nové bludiště a zobrazí jej místo starého.
7. Systém vykoná UC18 Vytvořit žebříček.

Alternativní tok:

- 4.1 Uživatel zruší svůj pokyn k vytvoření nového bludiště a tento use-case je tímto ukončen.
- 5.1 Zadané hodnoty nejsou validní a systém je nahradí výchozími hodnotami.

Podmínky pro dokončení: Do databáze byl vložen nový kompletní graf a uživatelům je dostupný nový žebříček.

UC18 – Vytvořit žebříček

Systém umožňuje tvorbu dalších žebříčků.

Aktéři: Systém.

Podmínky pro spuštění: Systém obdrží informaci o nově vytvářeném bludišti.

Základní tok:

1. Systém vytvoří žebříček pro dané bludiště.

2.2 Návrh

Vzhledem k poslání a zaměření experimentu je jeho podoba nasnadě. Z důvodu snadného přístupu a umožnění hraní co největšímu počtu hráčů se v podstatě musí jednat o hru hratelnou po Internetu. Zároveň lze poměrně intuitivně dospět i k tomu, že nejlepším zpodobněním problému obchodního cestujícího je bludiště. Uzly grafu budou reprezentovány jako místnosti a hrany jako chodby. Váha každé hrany pak může být specifikována jako nějaká forma překážky. Procházení labyrinty je navíc populární již odnepaměti a dle zaměření herních vývojářů (vizte Graf 5) se těmto hrám trápícím naše hlavy daří i nadále.

2.2.1 Platforma

Existuje spousta her, které se snaží vytěžit maximum z potenciálu lidského mozku k dosažení něčeho užitečného (vizte 1.4). Lidé stále předčí počítače v mnoha úkonech – např. v rozpoznávání objektů, jako jsou třeba galaxie (Galaxy Zoo⁸). Zatím však žádný z těchto projektů ani zdaleka nedosáhl popularity řekněme Minecraftu⁹, který od roku svého vydání (2009) hlásí již 107,86 milionů prodaných kopií a každým dnem tato cifra narůstá o dalších 53 000 (31). S tak ohromným záběrem by gamesourcing dokázal vskutku zajímavé věci, ale bohužel se takové oblibě žádné dosud existující crowdsourcingové hře nepodařilo dosáhnout.

Takže k návrhu experimentu jsem přistupoval tímto způsobem: nesnažit se vymyslet hru, která by řešila nějaký problém, a následně se ji pokusit podat tak, aby byla co nejzábavnější a populární, ale místo toho se pokusit vzít hru, která již svou popularitu získala a upravit ji tak, aby náš problém začala řešit.

Jen si to představme – mohlo by spojení všech těch hráčů, kteří hrají střílečky jako Call of Duty¹⁰ nebo strategické hry jako Starcraft¹¹ pomoci v léčení rakoviny? Mohlo by „farmerit“ zlatěáků, které je tak klíčové ve World of Warcraft¹² nebo sociální hry typu Farmville¹³ dělat něco užitečného? Spousta vývojářů by řekla, že ne – hraní je zábavné tím, že se nejedná o práci. Existují však v těchto hrách prvky, které jsou svou formou pro počítač výpočetně náročné, ale pro člověka snadné? Pokud by vědci dokázali na tuto otázku odpovědět, měli by mocný nástroj k řešení velkých problémů. Nutno podotknout, že i z hlediska finančního profitu by jistě šlo o zajímavou věc.

Já takové prvky vidím například v již zmiňovaném Minecraftu, kde krumpáčem těžíte materiál na stavbu nástrojů a budov, abyste měli kde přechkat noci plné nebezpečných tvorů. Jedná se o jednoduchý koncept, ale existuje nespočet variant a všelijakých módů. Vše je navíc vytvořeno z kostiček, což přímo vyzývá k snadné tvorbě třeba právě bludišť. Dokážu si i představit mód, kde by bylo nutné budovat proteiny ve stylu Folditu (vizte 1.4.3) nebo postavit raketu a následně s ní prolétávat vesmírem plným asteroidů jako v Genes in Space (vizte 1.4.4). Bohužel já sám jsem Minecraft nikdy nehrál a nedisponuji zkušenostmi s adekvátními vývojovými nástroji, takže vývoj takového experimentu by byl velmi

⁸ Dostupné na: <https://www.galaxyzoo.org/>

⁹ Více na: <https://minecraft.net/en-us/>

¹⁰ Více na: <https://www.callofduty.com/>

¹¹ Více na: <http://us.battle.net/sc2/en/>

¹² Více na: <https://worldofwarcraft.com/en-us/>

¹³ Dostupné na: <https://www.facebook.com/FarmVille/>

(časově) náročný. Nicméně potenciál z hlediska gamesourcingu je zde obrovský a dle mého názoru je jen otázkou času, kdy se této myšlenky někdo chopí. Zatím se objevila pouze snaha o využití s edukativními účely (32).

Zůstal jsem však u svého plánu zakomponovat experiment do již existujícího a fungujícího herního prostředí. Už 9 let funguji jako vývojář a programátor online prohlížečové hry Immortal Fighters a rozhodl jsem se toho využít.

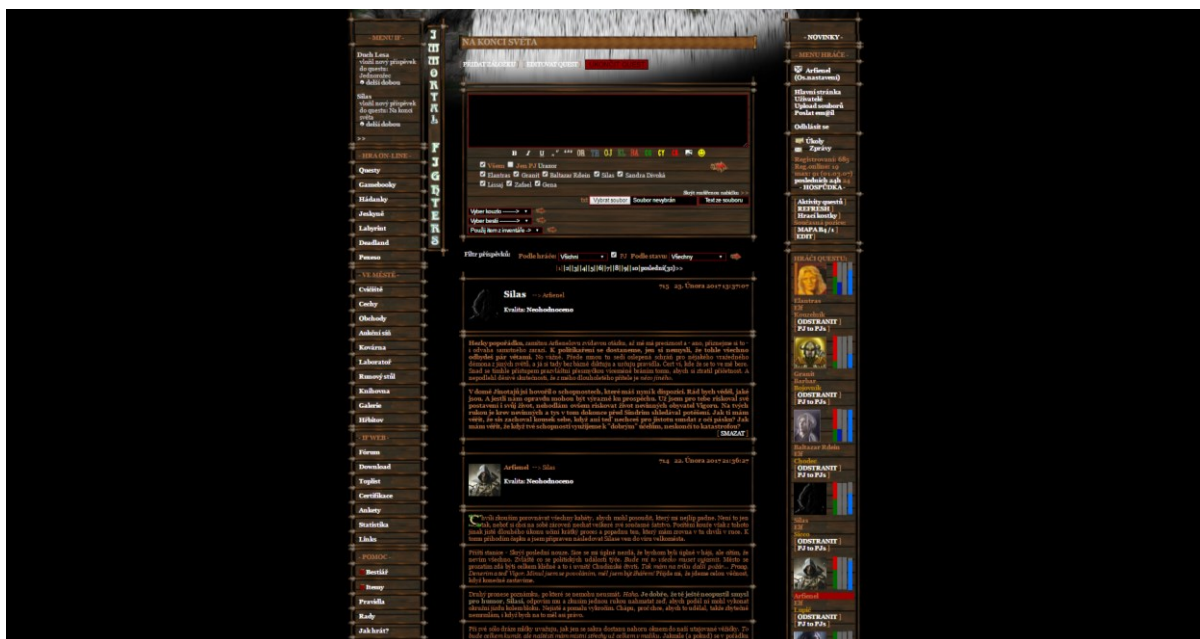
Immortal Fighters

Jedná se o portál dostupný na adrese <http://www.immortalfighters.net/>, který umožňuje hrát Dračí doupe online. Dračí doupe je legendární česká hra na hrdiny, která se inspirovala v systému Dungeons & Dragons¹⁴. Hru vede Pán, či Paní jeskyně (zkráceně PJ). Tento „vypravěč dobrodružství“ zná pravidla hry a ve své fantazii vytvoří svět, ve kterém hrají ostatní hráči prostřednictvím svých vymyšlených postav. Hra zpravidla probíhá tak, že PJ popíše, co se děje, a hráči řeknou, jak na to reaguje jejich postava. PJ se řídí mechanismy popsány v pravidlech, a i když má ve hře hlavní slovo, často si hází kostkami a používá údaje (sílu, obratnost, inteligence, charisma, atd.), které má každá postava různé v závislosti na rase (člověk, elf, trpaslík, atd.) a povolání (válečník, kouzelník, hraničář, atd.).

Dračí doupe je RPG neboli role-playing game: hra na hrdiny. Role-play znamená, že se hráč vžívá do vymyšlené postavy a hraje za ni, jako by byla živá. Hráči často do postav promítají svou vlastní osobnost, je však důležité si uvědomit, že hráč není postava a že je nutné je oddělovat. Při hraní za svou postavu se musí řídit nejen jejími vlastnostmi (zejména inteligencí a charismatem), ale také oddělovat své vědomosti od vědomostí své postavy (ta například nemůže vědět, že je někdo povoláním zloděj). Nejlepší hráči DrD nejsou ti, jejichž postavy zabijí nejvíce příšer, ale ti, kteří hrají nejlepší RP.

Zatímco klasické Dračí doupe vyžaduje, aby se hráči i PJ fyzicky sešli a příběh nahlas vyprávěli, online verze je založená na psaní. Oproti offline verzi chybí přímý osobní kontakt a hra je také mnohem pomalejší – více připomíná sdílené psaní povídky či románu. Hra probíhá tak, že PJ napíše příspěvek, ve kterém detailně představí situaci. Hráči pak ve vlastních příspěvcích popisují nejen to, jak jejich postava reaguje, ale také co si myslí, jak se cítí a jak vnímá okolní svět. DrD online tedy umožňuje mnohem propracovanější RP hru, jde více do hloubky. Navíc jej lze hrát odkudkoliv a příspěvek hráč nemusí psát v přesně určeném čase.

¹⁴ Více na: <http://dnd.wizards.com/>



Obrázek 22 – Pohled na klíčovou část webu Immortal Fighters – jeden z rozehraných příběhů

Krom toho na portálu Immortal Fighters existuje ještě množství modulů (kovárna, jeskyně, hádanky a další) sloužících pouze jako doplnění hlavního účelu hraní příběhů. Nejsou na příběhy nijak navázané a fungují paralelně. Využívají se k trávení času při čekání na příspěvky ostatních hráčů či PJ v příběhu. Ve většině z nich (za účelem vyšší motivace) lze i získávat herní měnu (zlaťáky, slouží k nákupu vybavení, zbraní atp.) nebo zkušenosti (slouží k navyšování úrovně postavy, čímž získává nové schopnosti, více životů, energie atp.).

Můj Labyrint se měl stát právě jedním z takovýchto rozšíření.

V současnosti aktivně hraje na portálu několik stovek hráčů z různých koutů ČR a Slovenska, což skýtalo poměrně slušnou startovací základnu pro rozběhnutí tohoto experimentu.

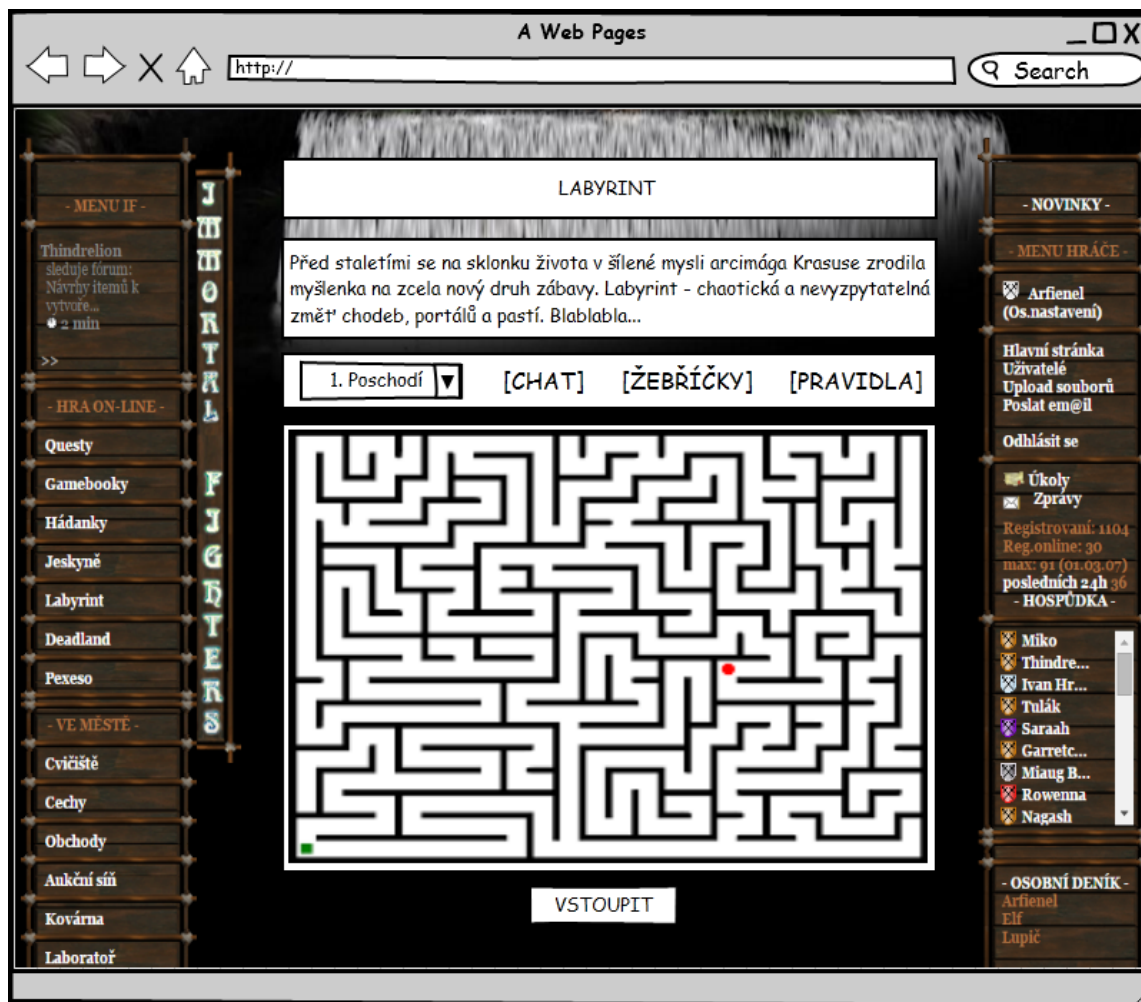
2.2.2 Vzhled

Uživatelské rozhraní IF sestává ze tří částí.

1. **Levé menu** slouží jako hlavní rozcestník a obsahuje odkazy na veškeré moduly, fórum, FAQ, atd. Zde jsem umístil i odkaz na Labyrint. Umístěn je v horní sekci menu, kde se vyskytují nejvýznamnější „volnočasové“ moduly v čele s odkazem na seznam aktuálně rozehraných DrD her (hlavní stavební kámen celého IF portálu). Experiment zde bude zároveň nejvíce na očích a snáze přiláká pozornost i nově registrovaných hráčů bez nutnosti jiné významnější propagace nebo upozorňování.
2. **Pravé menu** je z velké části individualizováno na základě hráčovy postavy. V horní části se nachází prvky pro odhlášení (popř. přihlášení, pokud uživatel není přihlášen), úpravy herního nastavení a nastavení účtu. Pokud je hráč přihlášen, je zde potom i seznam aktuálně přihlášených hráčů a informace o vlastní herní postavě. V tomto menu nebylo třeba pro mé účely nic měnit.
3. **Obsah** je centrální oblast mezi oběma menu, kde se zobrazují příslušné webové stránky dostupné z jednoho z nich. Při nutnosti dalšího strukturálního členění bývá v některých

případech při horní hranici této sekce umístěno další menu. Toho využívá pro větší přehlednost i Labyrint.

Prvotní koncept designu Labyrintu po klepnutí na odkaz Labyrint v levém menu je naznačen na obrázku dole:



Obrázek 23 – Prvotní designový koncept Labyrintu

Krom **nadpisu**, kterým je na portálu každý modul uveden, je na úvodní stránce Labyrintu přítomna i krátká **průpovídka**, jež celý tento nový modul zasadí do fantasy prostředí DrD. To především z důvodu, aby se ospravedlnilo následné získávání odměn, jako jsou zkušenosti nebo zlatáky, které posléze mohou hráči pomoci v hlavním RPG modulu.

Přítomno je dříve již zmiňované **menu**, které obsahuje rozbalovací nabídku pro **výběr podlaží** bludiště (důvod vysvětlen dále) spolu s odkazy na **chat**, **žebříčky** a **pravidla**. V případě, že se uživatel nachází na jiné než úvodní stránce Labyrintu, je výběr podlaží nahrazen odkazem **Bludiště**, kterým se na tuto stránku dostane.

Následuje podrobný popis jednotlivých stránek obsažených v rámci modulu Labyrintu a jednotlivé případy užití, které realizují.

1. Úvodní stránka (Bludiště)

Klíčovým prvkem je grafická reprezentace bludiště. Probíhá zde hlavní činnosti hry. Vstup do bludiště, pohyb po bludišti, zobrazování poloh ostatních hráčů, odchod z bludiště. Přítomny jsou rovněž informace o ztrátách herních zdrojů v důsledku pohybu (životů, staminy) a o skóre a jeho změnách.

V menu je přítomna rozbalovací nabídka pro výběr podlaží. Jelikož bludiště bude vždy reprezentovat úplný graf, nebylo by jej možné (při vyšším počtu místností) vykreslit na jedné rovině bez křížení chodeb. Na tom by nemuselo být nic až tak špatného, ale dle mého názoru by to působilo zmatečně a nepřehledně. Rovinný graf (tj. ten, který lze v rovině nakreslit bez křížících se hran) je limitován pravidly stanovenými v následující větě (33).

Věta (Eulerova formule): *V rovinném grafu $G = (V, E)$ platí:*

1. $|V| + |S| = |E| + 2$,
2. $|E| \leq 3|V| - 6$.

V jsou vrcholy, E hrany a S stěny grafu. Už graf K5 (10 hran) tedy není rovinný. A Labyrint bude pracovat i s mnohem většími grafy. Tudíž jsem přišel s řešením v podobě podlaží. Každá etáž zobrazí pouze tolik hran, aby bylo možno je vykreslit bez překřížení. Místnosti jsou přitom na každém poschodí umístěny na stejné pozici. V kontextu bludiště si toto řešení lze např. představit tak, že každá místnost obsahuje točité schodiště a na něm vchody do chodeb na příslušných podlažích.

Co se týče klíčových interakcí s bludištěm – vstup, pohyb, výstup – ty jsou řešeny pomocí dynamických oken, která se objeví při přejetí myši přes místnost. Každé takové okno obsahuje jak příslušné tlačítko pro vstup/pohyb, tak i další dodatečné informace. Kupříkladu právě přítomné hráče nebo překážku, která se nachází v chodbě do dané místnosti.

Pro admina je zde k dispozici ještě formulář pro tvorbu a spouštění nových bludišť a jejich uzavírání/otevírání (např. z důvodu oprav nebo testování nových funkcionalit).

2. Chat

Chat má formu diskuzního vlákna, tak jak jsou na to hráči IF již zvyklí. Sloužit má především ke komunikaci mezi hráči a adminy. Na hlášení chyb a navrhování vylepšení a dotazy ohledně Labyrintu. Práva mazat příspěvky přísluší pouze adminům, editace smí provádět jedině autor příspěvku. Samozřejmostí je i stránkování (po deseti).

3. Žebříčky

Stránka s žebříčky umožňuje listovat žebříčky všech proběhnuvších bludišť, ale především i toho aktuálního. Každý seznam obsahuje všechny odehrané pokusy všech hráčů seřazené podle skóre od nejlepšího po nejhorší. Je možné se prokliknout ke každé dané trase a prohlédnout si ji. Taky jsou zde definovány odměny.

4. Pravidla

Asi netřeba nic moc dodávat. K nalezení jsou tady instrukce, rady a tipy. A především informace co se na pozadí celé této hry vlastně odehrává.

5. Sledování trasy

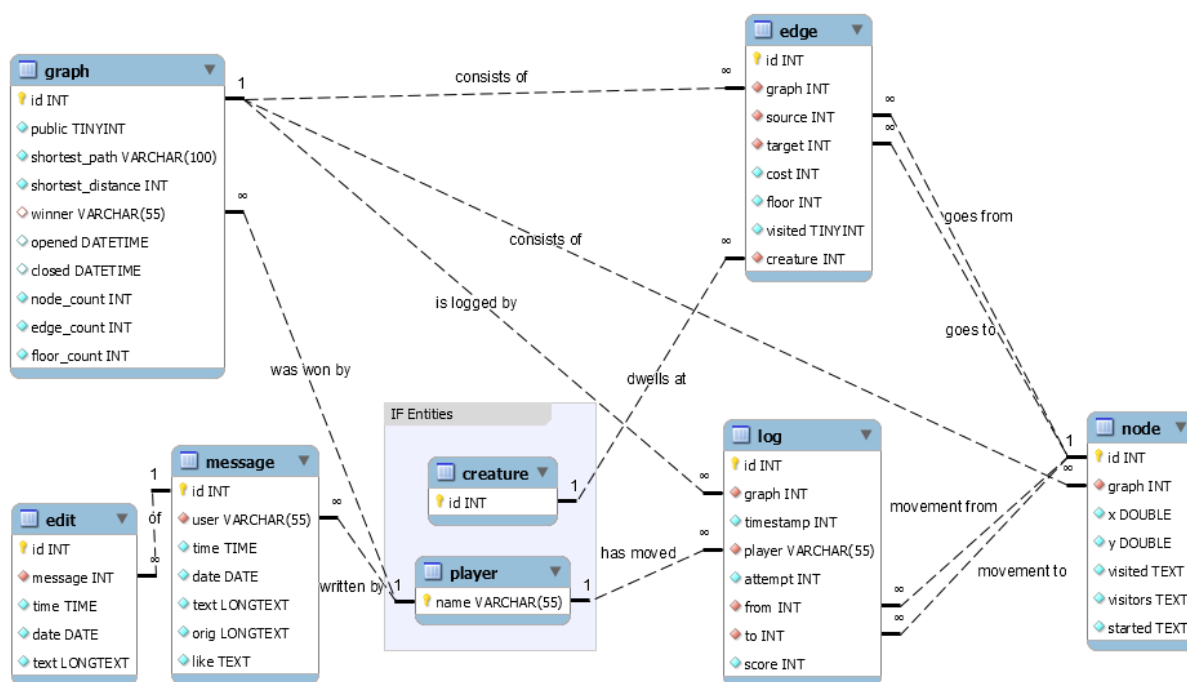
Jedná se o stránku přístupnou buďto skrze žebříček, nebo přes detail hráče nacházejícího se v některé z místností bludiště a zobrazuje trasy tohoto hráče. Tento prvek je velmi důležitý. Jedná se o adaptaci mravenčích feromonů. „Hodně lidí chodí tudy, tak já to taky zkusím.“

2.3 Implementace

Aplikace Labyrint byla vyvíjena dle třívrstvé architektury. Podkapitoly níže obsahují detailnější popis každé z vrstev, tedy – datové (databáze), logické (backend) a prezentační (frontend) plus místy pár ukázek či zajímavostí z kódu.

2.3.1 Databáze

Databáze portálu Immortal Fighters běží na MySQL serveru (ve verzi 5.0.68), tudíž jsem se ji rozhodl využít i pro Labyrint. Celkem jsem definoval 6 nových entit, které v databázi musely vzniknout jako nové tabulky. Ze stávajících entit IF portálu jsem využil pouze 2 (*creature* a *player*). Všechny, včetně vzájemných vazeb, jsou zachyceny na modelu Obrázek 24), kde primárním klíčem je vždy unikátní číselný identifikátor *id* s vlastností *AUTO_INCREMENT*, tedy automatického navyšování při vkládání nových záznamů. Bronzovou barvou jsou vyznačeny cizí klíče, modrou obyčejné atributy. Pokud je výplň prázdná, jedná se o atributy, jenž mohou být nulové (*NULL*). Všechny vazby jsou typu 1:N. Identifikoval jsem i několik vazeb M:N, ale pro povahu dat v jejich rámci, jež je vcelku dočasná a nestálá, jsem se rozhodl je nereprezentovat vazebními entitami, nýbrž formou řetězcových atributů (vysvětleno dále u konkrétních případů).



Obrázek 24 – Model databáze Labyrintu

Následuje významový a účelový popis jednotlivých entit a jejich atributů.

Entita graph

Reprezentuje graf a potažmo bludiště.

<u>Id:</u>	unikátní číselný identifikátor.
<u>Public:</u>	booleovská hodnota, která říká, jestli je bludiště právě teď přístupno hráčům. Výchozí hodnota je <i>false</i> .
<u>Shortest-path:</u>	řetězec obsahující nejkratší hamiltonovský cyklus v grafu ve formátu id uzlů oddělených pomlčkami. Pokud řešení není známo, zůstává toto pole prázdné.
<u>Shortest-distance:</u>	cena nejlevnějšího hamiltonovského cyklu.
<u>Winner:</u>	vítěz bludiště. Cizí klíč loginu hráče z entity <i>player</i> , který jako první našel ideální hamiltonovský cyklus v grafu.
<u>Opened:</u>	datum a čas spuštění bludiště, resp. vytvoření grafu.
<u>Closed:</u>	datum a čas ukončení bludiště. Ať už z důvodu nalezení správného řešení hráčem, anebo manuálního uzavření administrátorem.
<u>Node-count:</u>	počet uzlů v grafu., resp. místností v bludišti.
<u>Edge-count:</u>	počet hran v grafu, resp. chodeb v bludišti.
<u>Floor-count:</u>	počet pater v bludišti.

Entita node

Reprezentuje uzel grafu, potažmo místnost bludiště.

<u>Id:</u>	unikátní číselný identifikátor.
<u>Graph:</u>	cizí klíč identifikátoru grafu z entity <i>graph</i> , jemuž uzel přísluší.
<u>X:</u>	horizontální pozice pro renderování. Reálné číslo v rozmezí 0 až 1.
<u>Y:</u>	vertikální pozice pro renderování. Reálné číslo v rozmezí 0 až 1.
<u>Visited:</u>	realizace M:N vazby s entitou <i>player</i> ve formě řetězce obsahujícího loginy hráčů, kteří již tímto uzlem (resp. místností) prošli, oddělené středníky.
<u>Visitors:</u>	realizace M:N vazby s entitou <i>player</i> ve formě řetězce obsahujícího loginy hráčů, kteří se právě v uzlu (resp. místnosti) nacházejí, oddělené středníky.
<u>Started:</u>	realizace M:N vazby s entitou <i>player</i> ve formě řetězce obsahujícího loginy hráčů, pro které je tento uzel výchozí (započali v něm pohyb po bludišti), oddělené středníky.

Entita edge

Reprezentuje hranu grafu a chodbu bludiště.

<u>Id:</u>	unikátní číselný identifikátor.
<u>Graph:</u>	cizí klíč identifikátoru grafu z entity <i>graph</i> , jemuž hrana přísluší.
<u>Source:</u>	cizí klíč identifikátoru uzlu z entity <i>node</i> , ze kterého hrana vede.
<u>Target:</u>	cizí klíč identifikátoru uzlu z entity <i>node</i> , do kterého hrana vede.
<u>Cost:</u>	celočíselná váha (cena, vzdálenost) hrany. Nejméně 1.
<u>Floor:</u>	podlaží bludiště, na kterém se chodba nachází.
<u>Visited:</u>	booleovská hodnota určující, zdali tuto hranu (resp. chodbu) již někdo použil. Výchozí na <i>false</i> .
<u>Creature:</u>	cizí klíč identifikátoru příšery z entity <i>creature</i> , která v chodbě číhá.

Entita log

Reprezentuje informační záznam o akcích (pohybech) hráče v bludišti.

<u>Id:</u>	unikátní číselný identifikátor.
<u>Graph:</u>	cizí klíč identifikátoru grafu z entity <i>graph</i> . Říká, v rámci kterého grafu byl proveden pohyb.
<u>Timestamp:</u>	unixový čas v momentě provedení pohybu, resp. vytvoření logu.
<u>Player:</u>	cizí klíč loginu hráče z entity <i>player</i> , který vykonal pohyb.
<u>Attempt:</u>	číslo pokusu hráče. Tzn. o kolikátý průchod daným grafem se hráč pokouší.
<u>From:</u>	cizí klíč identifikátoru uzlu z entity <i>node</i> určující výchozí bod pohybu.
<u>To:</u>	cizí klíč identifikátoru uzlu z entity <i>node</i> určující cílový bod pohybu.
<u>Score:</u>	celkové skóre dosažené po vykonání pohybu.

Entita creature

Reprezentuje příšeru. Jedná se o entitu z IF portálu, z důvodu velkého množství atributů uvedu jen PK.

<u>Id:</u>	unikátní číselný identifikátor.
-------------------	---------------------------------

Entita message

Reprezentuje zprávu v chatu.

<u>Id:</u>	unikátní číselný identifikátor.
<u>User:</u>	cizí klíč loginu hráče z entity <i>player</i> , který je odesilatelem zprávy.
<u>Time:</u>	čas odeslání zprávy.
<u>Date:</u>	datum odeslání zprávy.
<u>Text:</u>	již naformátovaný text zprávy (vč. HTML tagů, zalomení řádku atd.).

- Orig: původní text zprávy tak, jak jej uživatel napsal do textového pole formuláře.
- Like: realizace M:N vazby s entitou *player* ve formě řetězce obsahujícího loginy hráčů, kteří této zprávě dali lajk.

Entita edit

Reprezentuje zprávu před editací.

- Id: unikátní číselný identifikátor.
- Message: cizí klíč identifikátoru zprávy z entity *message*, která byla editována.
- Time: čas editace.
- Date: datum editace.
- Text: již naformátovaný text původní zprávy (vč. HTML tagů, zalomení řádku atd.).

Entita player

Reprezentuje hráče. Jedná se o entitu z IF portálu, z důvodu velkého množství atributů uvedu jen PK.

- Name: unikátní login hráče.

2.3.2 Backend

Za backend považuji veškerou činnost prováděnou serverem. Tzn. zprostředkování dat z databáze klientovi a naopak úprava dat v databázi na základě uživatelských akcí plus backend samozřejmě zajišťuje i hlavní funkcionality aplikace. Jelikož Immortal Fighters běží na Apache serveru s PHP 5.2.6, tak i backend jsem programoval v tomto jazyce. Nevyužíval jsem příliš jeho objektových rysů, protože jsem zvyklý spíše na jeho procedurální podobu. Objektově jsem tedy programoval jen výjimečně.

Přenos dat

Získávání dat a jejich přenos do frontendu probíhá skrze data-transfer objekty. Na pokyn klienta jsou z databáze dotazem vybrány požadované údaje, a ty se následně uloží do objektu, který nemá žádnou jinou funkcionalitu. Slouží pouze jako struktura pro přenos. Tento objekt je potom převeden do formátu JSON a předán klientovi, který si jej už potom zpracuje a využije k obrazu svému. Níže je například skript pro získání všech uzlů z aktuálního grafu, respektive místností z bludiště.

```
class Node {
    public $id;
    public $x;
    public $y;
    public $wasVisited;
    public $isActual;
    public $isStarting;
}
$result = array();
$graphId = $database->query("SELECT `id` FROM `graph` ORDER BY `id` DESC LIMIT 1")->fetchColumn();
$nodes = $database->query("SELECT * FROM `node` WHERE `graph` = '$graphId'")->fetchAll();
foreach ($nodes as $node) {
    $item = new Node();
    $item->id = $node["id"];
    $item->x = $node["x"];
    $item->y = $node["y"];
}
```

```

$item->wasVisited = strpos($node["visited"], $_SESSION["auth_user_session"]) !== false;
$item->isActual = strpos($node["visitors"], $_SESSION["auth_user_session"]) !== false;
$started = explode(";", $node["started"]);
$started = !empty($node["started"]) AND count($starters) > 0 ? $started : null;
$item->isStarting = strpos($node["started"], $_SESSION["auth_user_session"]) !== false;
array_push($result, $item);
}
echo json_encode($result);

```

Na začátku se nachází definice třídy, jejíž instance následně poslouží jako schránka na data. Následují dotazy na databázi. První, který vybere *id* aktuálního grafu (tzn. ten s maximálním *id*) a druhý, jenž za pomoci onoho *id* vybere veškeré uzly příslušející danému grafu a uloží je ve formě dvoudimenzionálního pole. To je poté proiterováno v cyklu, kde pro každý uzel vznikne jeho data-transfer objekt třídy *Node*. Všechny objekty jsou vloženy do nového pole, které je převedeno do řetězce JSON a ten je výstupem pro klienta (frontend).

Analogickým způsobem je řešen i opačný postup, tedy přenos dat od klienta na server.

Tvorba bludiště

Labyrint umožňuje 2 způsoby, jak vytvořit nové bludiště.

1. **Náhodně** – admin pouze zadá počet místností a podlaží a systém se již postará o zbytek.
2. **Dle souboru** – admin vybere TXT soubor obsahující matici sousednosti (řádkovanou), specifikuje oddělovač jednotlivých hodnot, zadá počet podlaží a délku nejkratšího hamiltonovského cyklu.

Po pokynu k vytvoření nového bludiště se vytvoří nový graf o požadovaném počtu vrcholů. Každý vrchol získá náhodné souřadnice *x* a *y* v rozmezí 0 až 1. Mezi vrcholy dojde k vytvoření tolika hran, aby se jednalo o graf úplný, tedy:

$$|E| = \frac{|V|}{2} (1 + |V|) - |V| = \frac{|V|(|V|-1)}{2},$$

kde *V* jsou vrcholy a *E* hrany. Následně je každé hraně náhodně přiděleno podlaží v adminem stanoveném rozmezí, poté je zaznamenán nejkratší hamiltonovský cyklus (pokud byla tato informace předána), jeho vzdálenost (povinný údaj kvůli výpočtům skóre), a nakonec dojde na distribuci vah a příšer.

V případě náhodné tvorby bludiště se nejkratší cesta vytvoří uměle tak, že se zamíchá posloupnost všech uzlů grafu. Příslušným hranám na této posloupnosti je přiřazena váha 1, těm zbývajícím v grafu pak náhodná vyšší, limitovaná maximálním počtem životů (vizte dále).

V případě tvorby ze souboru je každé hraně přiřazena váha dle matice sousednosti.

Rozmístění monster je poslední a patrně nedůležitější fází tvorby bludiště, jelikož se jedná o údaj sdělující hráčům přibližnou cenu každé chodby a na základě této informace se rozhodují, kudy půjdou dál.

V databázi IF se nachází tabulka *creatures*, která obsahuje všechny stvůry používané ve fantasy světě tohoto portálu. Každá tato bytost má atribut nazvaný životaschopnost, který určuje, kolik životů může mít. Hodnota životaschopnosti vychází z pravidel DrD a říká, kolikrát je třeba hodit kostkou. Výsledek se sečte a vyjde počet životů, které daná příšera má. Obecně se dá říct, že platí vztah: čím více životů,

tím silnější nestvůra je. Systém tedy všechny bytosti vybere, zpracuje jejich životaschopnost (která je uložena ve formě řetězce) a „kostkami“ každému druhu nestvůry nahází životy. Příklady možných forem životaschopnosti ukazuje tabulka (Tabulka 1) níže.

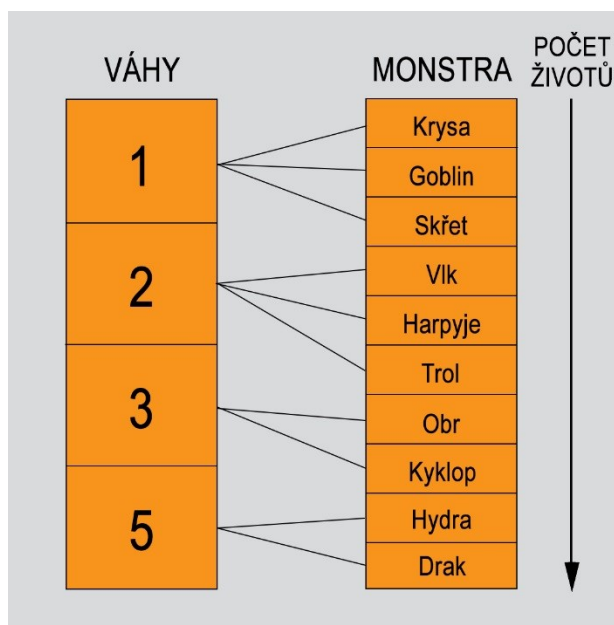
Tabulka 1 – Výklad životaschopnosti nestvůr

Životaschopnost	Počet životů
2	2krát hod 10stěnnou kostkou
1 + 2	1 hod 10stěnnou kostkou + 2
k8 - 1	1 hod 8stěnnou kostkou - 1
35-3	35 hodů 10stěnnou kostkou - 3
k6	1 hod 6stěnnou kostkou
2+1	2 hody 10stěnnou kostkou + 1

Jak je vidět, formát je to různorodý a není příliš jednotný. Např. čísla by od znamének měla být oddělena mezerami, ale v mnoha případech tomu tak není. Některá monstra mají dokonce životaschopnost zadánu formou textu (např. „3 mládě, 5 dospělý“), s takovými jsem se rozhodl nepracovat. Následujícím regulárním výrazem (dle MySQL syntaxe) lze z databáze vybrat všechny druhy příšer, které mají životaschopnost podobnou těm uvedeným v tabulce výše (Tabulka 1). Což činí cca 87 % ze všech v databázi.

```
^[k]?[1-9][0-9]*[:blank:]?[+-]?[:blank:]?[0-9]*$
```

Po vypočtení životů se dle nich všechny příšery vzestupně seřadí. Stejně tak jsou seřazeny i seskupené váhy hran v grafu. Každé jedné váhové kategorii je posléze přiřazeno tolik monster, aby jich každá kategorie měla ve výsledku pokud možno stejný počet. Přehledněji je to vysvětleno na obrázku níže.



Obrázek 25 – Příklad distribuce monster dle vah hran v grafu

Každé hraně s příslušnou vahou se poté náhodně vybere příšera z těch, které této váze přísluší. Tímto celým procesem je zajištěno, že i v případě, kdy by hráči vždy řešili dokola ten samý graf, budou na

stejných hranách různé příšery, a tak nebudou moci vycházet z informací získaných při procházení předchozích bludišť. Např. taktika „nejvýhodnější předtím bylo používat cesty se Skřetem, takže jimi půjdu i teď“ nemusí fungovat, protože onen Skřet v tomto bludišti získal jiný počet životů (padly jinak „kostky“), a tím pádem mohl být přiřazen k jiné váhové kategorii.

Veškeré výše zmíněné informace jsou následně uloženy do databáze na příslušných místech.

Pohyb bludištěm

Při vydání pokynu k pohybu dojde v první řadě ke kontrole, zdali má hráč dostatečný počet zdrojů pro pohyb. Těmito zdroji jsou životy a stamina (jakási výdrž či energie). Každý přesun z místnosti do místnosti totiž něco stojí. Pakliže je vše v pořádku a hráči je pohyb povolen, dostane se do druhé místnosti a je informován o změně svého skóre. Po každém vykonaném pohybu je do databáze vložen log s příslušnými fakty.

Skóre

Skóre má za úkol informovat hráče o tom, jak dobře si zatím vede. Výpočet tohoto údaje by neměl být na venek příliš průhledný, ale zároveň by měl přesně reflektovat potenciál aktuální hráčovy cesty stát se tou ideální.

V první fázi jsem experimentoval s nápadem, kdy by se skóre odvíjelo od aktuální délky trasy hráče vzhledem k té ideální s ohledem na provedený počet pohybů.

$$S(H, G) = \frac{|P_H|^2}{|V| * \sum w_G} * 1000$$

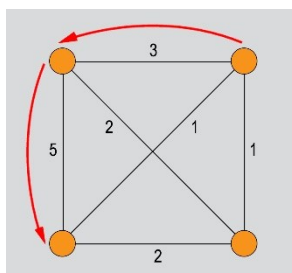
S je skóre, H hráč, G graf, P pohyb, V vrchol a w váha. Výhodou bylo, že pro každý graf se aktuální skóre hráče mohlo pohybovat v rozmezí od 0 do 1000, kdy 1000 znamenalo dosažení nejkratšího hamiltonovského cyklu. Tento způsob bodování byl použit při alfa testování, ale záhy se projevil jeho neduh. Zprvu, toto hodnocení nepočítalo s variantou, že by ideální cesta vedla po hranách s jinými vahami než 1 (v prvopočátku totiž nebylo možno vytvořit graf dle souboru). Zadruhé, tento systém bodování pozbýval jakéhokoliv smyslu z pohledu RP a zasazení do fantasy prostředí. Hráči získávali nějaké body, ale nedalo se to nijak odůvodnit připodobněním k jakémukoliv reálnému procesu.

Proto jsem se rozhodl skóre počítat jako prostý poměr sum hráčem navštívených vah a všech vah v grafu. Tím se význam skóre obrátil a méně začalo znamenat více. Okamžitě se tak nabídlo pracovat se skóre jako s časovým údajem říkajícím, kolik hráč zatím strávil v bludišti času chozením. To už je mnohem smysluplnější a pochopitelnější. Výsledný vzorec nakonec dosáhl následující podoby:

$$S(H, G) = \frac{\sum w_H}{\sum w_G} * 10^{\left\lceil \log \frac{w_G \min}{\sum w_G} \right\rceil + 1},$$

kde značení je stejné jako u předchozího vzorce ($w_G \min$ je nejkratší hrana grafu). Výsledek je v sekundách. Onen poměr součtů hráčem projitých vah a všech vah v grafu je transponován na základě nejlevnější váhy v grafu tak, aby nevycházela nesmyslně malá čísla (a tak třeba někdo neprošel chodbou např. za 0,0001 s). Teoreticky nejkratší možný přesun chodbou může tedy trvat 1 vteřinu. Tento výpočet skóre je již zcela univerzální a svou úlohu plní dokonale.

Na obrázku (Obrázek 26) je znázorněn příklad výpočtu skóre hráče, který se zatím na K4 grafu pohnul dvakrát. Celkem ušel vzdálenost 8. Součet všech vah v grafu je 14, nejkratší hrana má délku 1. Dle vztahu, který jsem definoval výše, tudíž hráčovo skóre v tento moment bude činit 57,14.



Obrázek 26 – Příklad pro výpočet aktuálního skóre

Chat

Implementace chatu je poměrně přímočará, jedná se o klasické pojetí diskuzního vlákna ve fóru.

Žebříčky

Backend vypíše všechny pokusy hráčů, kteří ve vybraném bludišti v rámci jednoho pokusu provedli alespoň jeden pohyb (a jejich skóre je tedy nenulové) ve vzestupném pořadí dle skóre (čím kratší čas hráči stačil na projití všemi místnostmi, tím lépe). Přitom mimo pořadí postaví výsledky, které ještě nejsou finální (hráč zatím nedokončil hamiltonovský cyklus) nebo jsou totožné s nějakou již dřívější trasou některého hráče (jsou duplicitní).

Pravidla

Zde se backend nijak nezapojuje, jedná se o čistě statickou stránku s přehledem pravidel hry.

Sledování tras

Zde backend figuruje pouze jako zprostředkovatel dat z databáze, na jejichž základě frontend vykreslí bludiště a znázorní trasu vybraného hráče.

2.3.3 Frontend

Za frontend pokládám kód vykonávaný klientem. Tzn. HTML5, CSS a JavaScript.

Bludiště

Klíčovým prvkem celého Labyrintu je samozřejmě bludiště. Rozhodoval jsem se mezi dvěmi alternativami pro jeho vykreslení. Buďto využít nového prvku představeného v HTML5 – Canvas, nebo formátu SVG. Canvas pracuje rastrově, SVG vektorově. Z hlediska výkonnosti má SVG navrch při větších rozměrech plátna, ale zase ztrácí při složitějších obrazcích. Z tohoto důvodu jsem se nejdříve přikláněl ke Canvas variantě. Jenže fakt, že Canvas vytváří pouze bitmapy, nakonec definitivně rozhodl ve prospěch SVG, nehledě na případné výkonnostní otazníky. SVG totiž funguje jako XML dokument a lze s jednotlivými vykreslenými prvky pracovat jako s objekty, které lze stylovat za pomoci CSS, přiřazovat jim události apod. Vektorová grafika rovněž umožňuje i pohodlný zoom, který najde své využití při složitějších bludištích, která by mohla na omezeném prostoru obrazovky už vypadat poněkud nepřehledně.

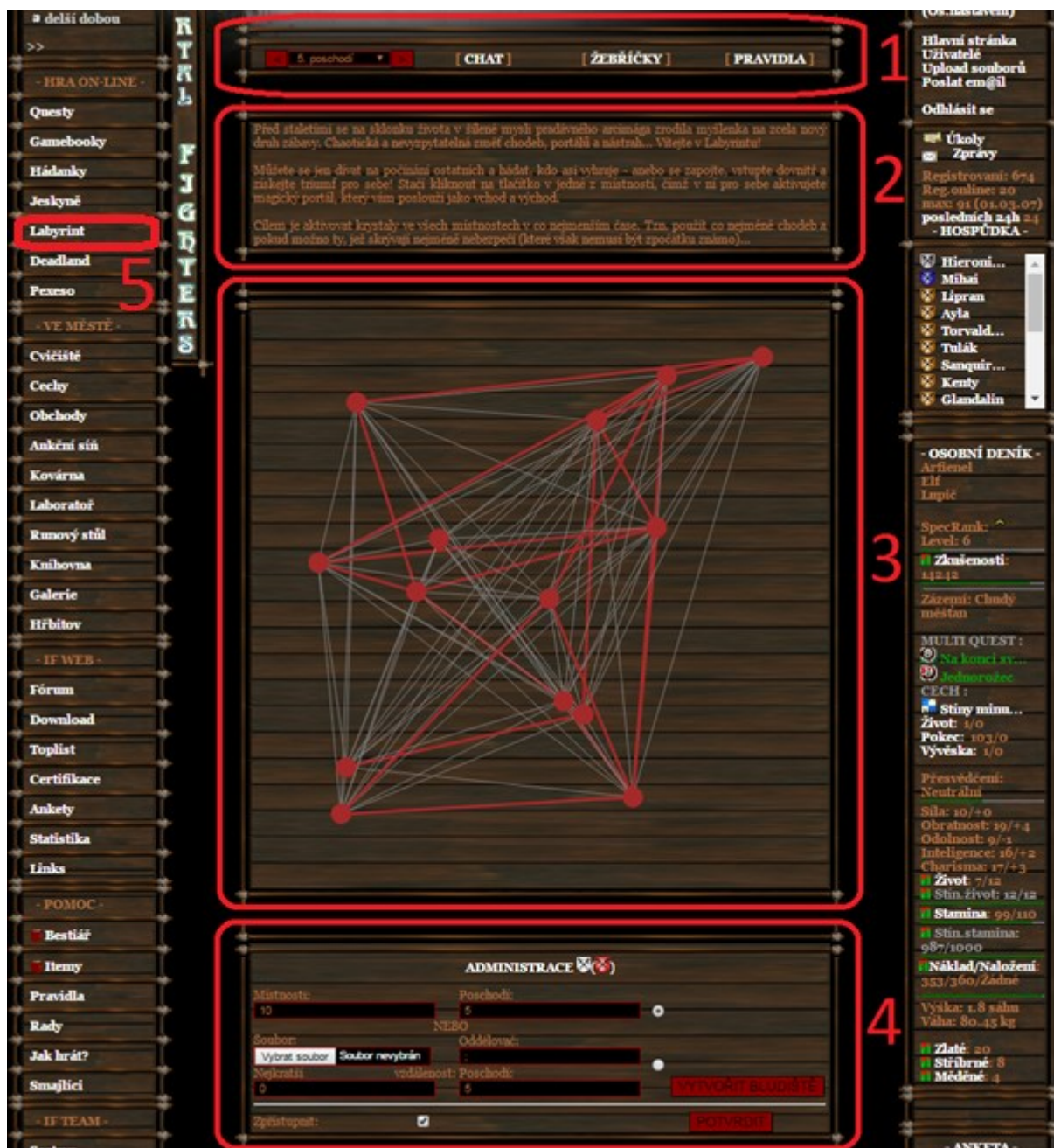
Obavy z výkonu se posléze ukázaly jako plané, struktura bludiště totiž není zas až tak komplikovaná, jak se na první pohled zdálo.

```
<svg id="mazeSvg" width="555" height="555" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <marker id="arrow" markerWidth="6" markerHeight="6" refX="9" refY="3" orient="auto">
      <path d="M0,0 L0,6 L6,3 z" fill="white"/>
    </marker>
    <marker id="arrow2" markerWidth="6" markerHeight="6" refX="-3" refY="3" orient="auto">
      <path d="M0,3 L6,6 L6,0 z" fill="white"/>
    </marker>
  </defs>
  <g>
    <line id="3795" stroke="brown" stroke-width="3" x1="300" y1="374" x2="367" y2="467"/>
    (...)
  </g>
  <g fill="brown">
    <circle r="10" cx="368" cy="468" id="644" pointer-events="all"/>
    (...)
  </g>
</svg>
```

Kořenový element `<svg>` obsahuje pouze tři prvky. Prvním je sekce `<defs>`, která definuje struktury, na něž se lze poté ve zbytku SVG dokumentu odkazovat skrze jejich *id*. Bludiště (zatím) obsahuje pouze dvě, a to typu `<marker>`, jejichž využití rozeberu dále. Další už je jen dvojice tagů `<g>`, které značí vrstvy. První (spodní) vrstva obsahuje hrany reprezentované prostými čarami `<line>`, druhá vrcholy, kterými jsou kruhy (`<circle>`), popřípadě čtverce (`<rect>`). O vytvoření této struktury se stará JavaScript, který z backendu získá informace o vrcholech a hranách. Vrcholy umístí na plátno na základě jejich x a y souřadnic, hrany pak vykreslí jako jejich spojnice. Barva vrcholů je stanovena na hnědou, hrany jsou hnědé, pokud se nacházejí na aktuálně zvoleném poschodí, jinak jsou znázorněny šedě. Všem vrcholům jsou dále k událostem `onmouseenter` a `onmouseleave` přiřazeny funkce pro zobrazení či schování okýnka obsahujícího detaily ohledně příslušné místnosti, hraně spojující aktuální místnost hráče a vybranou místnost a v neposlední řadě také možnost se do dané místnosti přesunout. Požadavky na backend jsou zasílány metodou AJAX skrze HTTP request typu POST. Kód celého skriptu naleznete na příloženém CD v souboru *maze.js*.

Na obrázku níže (Obrázek 25) je úvodní a hlavní stránka Labyrintu s červeně vyznačenými oblastmi.

1. **Menu** – umožňuje přepínat poschodí bludiště a přecházet na další součásti Labyrintu, tzn. chat, žebříčky a pravidla.
2. **Info** – rámeček zobrazující potřebné informace pro hru. Zasazuje bludiště do fantasy příběhu, navádí hráče, co má dělat a sděluje mu informace o průběhu jeho hry a skóre.
3. **Bludiště** – grafická reprezentace grafu. Skrze vyskakovací okýnka u každého uzlu je možné do bludiště vstoupit a následně jím procházet.
4. **Administrace** – dostupná pouze adminům, kteří odsud mohou vytvářet bludiště nová, a to buď dle matice sousednosti v TXT souboru, nebo náhodně. Rovněž lze aktuální bludiště uzavřít, či otevřít (např. pro případ údržby).
5. **Odkaz** – v levém menu portálu IF, přes který se lze na tuto stránku dostat.



Obrázek 27 – Hlavní stránka Labyrintu

Průběh hry

K zapojení do hry je nejprve nutné vstoupit do bludiště. To lze provést tak, že uživatel vybere jednu z místností (najede na ni myší), čímž se zobrazí okýnko obsahující detaily o dané místnosti (vizte níže Obrázek 28), především však tlačítko pro vstup. Po jeho stisknutí se vybraná místnost hráči začne zobrazovat jako čtverec, čímž je signalizován počáteční (a zároveň závěrečný) vrchol. Ve vyskakovacích okýnkách již nyní bude tlačítko pro přesun a budou k dispozici i informace o překážce v chodbě, která spojuje příslušnou místnost s místností aktuální. Aktuální místnost je vždy značena bíle. V informačním rámečku mezi menu a bludištěm se zobrazí barevná stupnice vyznačující aktuální hráčův čas v poměru k ideálnímu času.



Obrázek 28 – Detail vyskakovacího okýnka místnosti

Vyskakovací okýnko obsahuje následující informace:

1. **Přítomní hráči** – na obrázku místnosti jsou zobrazeny avatary všech hráčů přítomných v dané místnosti, kliknutím na některý z nich se otevře další záložka prohlížeče se sledováním trasy tohoto hráče.
2. **Krystal** – znázorňuje, zdali uživatel tuto místnost již navštívil. Pakliže ano, krystal září a rotuje.
3. **Nestvůra** – obrázek nestvůry, jež se nachází v chodbě mezi aktuální místností uživatele a vybranou místností. Napovídá, jak moc bude výhodné tuto chodbu použít (např. Drak asi uživatele zpomalí více než Krysa). Pokud tuto chodbu ještě nikdo nepoužil, pak nestvůra není známa.
4. **Portál** – znázorňuje vstupní bod hráčů. Je-li portál oranžový, jedná se o vstupní bod někoho jiného, pokud je modrý, jde o vstupní bod uživatele (a popř. i někoho dalšího). Zde daný hráč svou cestu bludištěm započal a musí ji zde také ukončit.
5. **Statistiky** – údaje o příslušné chodbě. Kolikrát již byla použita, jakého bylo dosaženo průměrného celkového času v trasách používajících tuto chodbu a tři nejlepší časy tras, které zahrnují tuto chodbu. Těmito statistickými informacemi, nestvůrou a sledováním tras ostatních by se hráči měli řídit, aby dosahovali co nejlepších výsledků.
6. **Tlačítko** – buď pro přesun do dané místnosti, nebo pro vstup do ní, v případě, že se uživatel v bludišti ještě nenachází. Pokud se příslušná chodba nachází na jiném poschodí, místo tlačítka se zobrazí text s údajem, na které poschodí je třeba přepnout.

Při najetí myši na jinou místnost rovněž dojde ke změně vizuální podoby chodby (krom barvy) z prosté čáry na šipku (jak je vidět na Obrázek 28). Toho je dosaženo za pomoci prvků <marker> zmíněných výše. Jedna ze dvou jeho předem definovaných variant, kdy v obou případech jde o trojúhelník jen jinak natočený, se (po rozhodnutí frontendu) aplikuje na konec prvku <line>, kterým je chodba reprezentována, a vznikne tak šipka.

Pokud se uživatel rozhodne do místnosti přesunout, dojde k výměně informací se serverem a znovunačtení stránky. Hráč se do cílového vrcholu přemístí, což je signalizováno i změnou barvy místnosti na bílou. Naopak opuštěná místnost se zabarví do tmavšího odstínu původní barvy, aby hráč

měl přehled o tom, kudy již prošel. Jestliže by uživatel neměl dostatek zdrojů (staminy, životů) pro pohyb, pak k přesunu nedojde a je o tomto faktu informován. Množství potřebných zdrojů pro jeden pohyb chodbou je stanoven fixně a slouží k motivaci pro pečlivější zvažování svých činů a prevenci před prostým, bezduchým klikáním (protože tyto zdroje se doplňují jen jedenkrát za 24 hodin a pouze o čtvrtinu). K přesunu rovněž nedojde, pokud se chtěl hráč dostat do jím již navštívené místnosti (krom výjimky při zakončování cyklu).

Během přesunu se přepočítá skóre (čas) a zapíše log. Změna ve skóre a informace o úbytku zdrojů se uživateli zobrazí v informačním rámečku nad bludištěm. Také se aktualizuje časová stupnice. Pokud celkový hráčův čas (a tedy délka jeho aktuální cesty) již přesáhl délku nejkratšího hamiltonovského cyklu, zabarví se stupnice do ruda a její maximální hodnotou se stane celkový součet všech délek hran v grafu.

Tímto způsobem hráč postupně navštíví všechny místnosti a následně je v informačním rámečku vyzván k návratu do výchozí místnosti (tedy té se svým portálem). Jakmile do ní přejde, je jeho cesta bludištěm ukončena a může začít celý proces od znovu (včetně vyvolání portálu).

Pokud uživatel v momentě po přesunu zpět do svého výchozího bodu dosáhne ideální délky trasy, bludiště se okamžitě pro všechny hráče uzavře. Potom je už na adminech, aby nechali vytvořit bludiště nové.

Chat

Stránka chatu je poměrně jednoduchá. Sestává z textového pole pro napsání zprávy s možností základní strukturalizace textu (tučné písmo, barva písma, vložení obrázku, smajlíku) a stránkovaného výpisu všech příspěvků (po deseti). Uživatel může následně svůj příspěvek upravovat, historie všech verzí upraveného příspěvku se potom komukoliv zobrazí po rozbalení spodní části příspěvku. Mazat příspěvky mohou pouze administrátoři. Celkově se jedná o klasický koncept diskuzního vlákna na fóru.



Obrázek 29 – Chat Labyrintu

Žebříčky

Na této stránce si mohou uživatelé prohlížet výsledky jednotlivých bludišť. Lze mezi nimi listovat podobným způsobem jako předtím měnit podlaží u bludiště – ovládacím prvkem v menu. Každý žebříček nejprve obsahuje informace o bludišti jako takovém – počet místností, chodeb a časové rozpětí od jeho otevření až po uzavření. Následují vysvětlivky symbolů, které se v žebříčku mohou objevit a textové pole pro filtr dle jména hráče.

Poté už je vypsaná tabulka výsledků pro dané bludiště, seřazená dle skóre (času) vzestupně, tedy od nejlepšího k nejhoršímu. Nejlépe je to vidět na obrázku níže (Obrázek 30).

BLUDIŠTĚ [CHAT] 2017-03-06 14 [PRAVIDLA]

Žebříček pro bludiště o 15 místnostech a 105 chodbách, rozvrstvených do 7 podlaží z období: 2017-03-06 14:48:19 až 2017-03-06 20:49:56.

🥇 = 1. místo 🥈 = 2. místo 🥉 = 3. místo 🏆 = 4. místo
! = duplikovaná trasa ... = (zatím) nedokončená trasa

Filtr: Jméno hrdiny...

Pořadí	Hrdna	Rasa	Povolání	Pokus	Čas	Trasa	Odměna
1.	Dawyx91	Elf	Kouzelník	16.	27	👁	...
2.	Caramiels	Elfka	Kouzelnice	8.	29	👁	...
3.	Silas	Elf	Sicco	5.	29	👁	🥇
4.	Dawyx91	Elf	Kouzelník	14.	42	👁	🥈
5.	Silas	Elf	Sicco	4.	42	👁	...
6.	Dawyx91	Elf	Kouzelník	6.	48	👁	...
7.	Dawyx91	Elf	Kouzelník	9.	48	👁	...
8.	Dawyx91	Elf	Kouzelník	12.	48	👁	...
9.	Dawyx91	Elf	Kouzelník	13.	48	👁	!
10.	Silas	Elf	Sicco	3.	52	👁	...
11.	Dawyx91	Elf	Kouzelník	15.	52	👁	...
12.	Dawyx91	Elf	Kouzelník	10.	54	👁	...
13.	Dawyx91	Elf	Kouzelník	8.	59	👁	...
14.	Dawyx91	Elf	Kouzelník	11.	63	👁	...
15.	Silas	Elf	Sicco	1.	63	👁	...
16.	Dawyx91	Elf	Kouzelník	7.	65	👁	...
17.	Caramiels	Elfka	Kouzelnice	5.	69	👁	🥉
18.	Caramiels	Elfka	Kouzelnice	6.	69	👁	...
19.	Dawyx91	Elf	Kouzelník	5.	73	👁	...
20.	Caramiels	Elfka	Kouzelnice	7.	73	👁	...
21.	Dawyx91	Elf	Kouzelník	4.	80	👁	...
22.	Silas	Elf	Sicco	2.	84	👁	...
23.	Arfiel	Elf	Lupič	4.	88	👁	🥇
24.	Dawyx91	Elf	Kouzelník	3.	90	👁	...
25.	Arfiel	Elf	Lupič	3.	90	👁	...
26.	Arfiel	Elf	Lupič	1.	105	👁	...
27.	Caramiels	Elfka	Kouzelnice	2.	111	👁	...
28.	Tanwar	Barbar	Válečník	2.	117	👁	...
29.	Dawyx91	Elf	Kouzelník	2.	117	👁	...
30.	Caramiels	Elfka	Kouzelnice	4.	120	👁	...
31.	Arfiel	Elf	Lupič	2.	126	👁	...
32.	Dawyx91	Elf	Kouzelník	1.	138	👁	...
33.	Caramiels	Elfka	Kouzelnice	1.	138	👁	...
34.	Caramiels	Elfka	Kouzelnice	3.	141	👁	...
35.	Tanwar	Barbar	Válečník	1.	178	👁	...

- OSOBNÍ DENÍK -
Arfiel
Elf
Lupič
SpecRank: ^
Level: 6
Zkušenosti: 14242
Zázemí: Chudý měšťan
MULTI QUEST: Na konci sv...
CECH: Stíny min...
Život: 1/0
Pokec: 103/0
Vývěska: 1/0
Presvědčení: Neutrální
Síla: 10/+0
Obratnost: 19/+4
Odolnost: 9/-1
Intelligence: 16/+2
Charisma: 17/+3
Život: 7/12
Stín Život: 12/12

Obrázek 30 – Žebříček Labyrintu

Zobrazen je každý pokus každého hráče, včetně těch (zatím) nedokončených. Ve sloupci *Odměna* je pak určeno, zdali se jedná o platný pokus a jaké je jeho faktické pořadí. První 4 umístění jsou označena zlatou až bramborovou medailí, přičemž jeden hráč může obsadit pouze jedinou pozici (takže v případě, že by někdo hrál zcela sám, u jeho nejlepšího pokusu by se zobrazovalo zlato, ostatní metály by chyběly).

Tři tečky značí neúplný pokus. To může být důsledkem toho, že onen hráč stále ještě bludištěm chodí (tzn., ještě jej neopustil portálem), nebo již někdo v bludišti našel nejlepší možný hamiltonovský cyklus, tím došlo k jeho uzavření, a ostatní už nemohli své pokusy dokončit.

Vykřičník znamená duplikovanou trasu. Tzn., že někdo zopakoval (ať už úmyslně, či nikoliv) trasu, která již byla zaznamenána předtím. Jedná se o ochranný prvek proti záměrnému kopírování kvalitních tras za účelem umístění se na vedoucích příčkách (druhé až čtvrté místo) a získání tak odměny v podstatě bez námahy. První 4 pozice jsou totiž na závěr odměněny herní měnou – zlaťáky.

Pokusy aktuálně přihlášeného uživatele jsou pro něj zvýrazněny červeně.

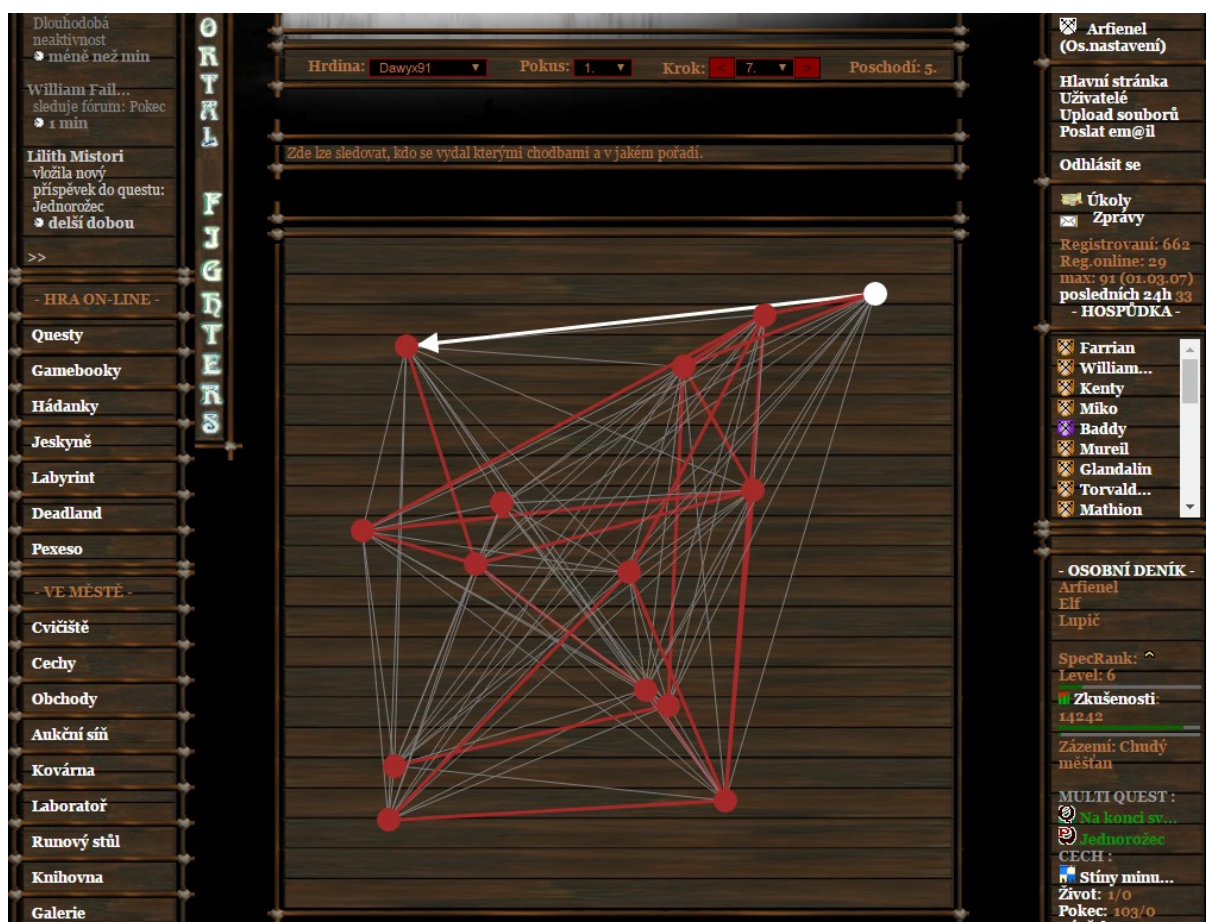
U každého pokusu je ikona oka, skrze kterou se lze prokliknout na stránku sledování tras, přičemž daná trasa bude rovnou vybrána.

Pravidla

Tato stránka je zcela statická a obsahuje informace důležité ke hraní. Po jejich přečtení by se měl být každý uživatel schopen okamžitě plnohodnotně zapojit do hry.

Sledování tras

Sem se lze dostat buďto skrze žebříček, nebo klikem na portrét hráče v místnosti bludiště.



Obrázek 31 – Sledování tras v Labyrintu

Sledování tras se otevře na nové záložce prohlížeče a horní menu je nahrazeno filtrovacími prvky. Lze vybrat hráče, jeho pokus a krok v daném pokusu. Ten se poté zobrazí v bludišti vykresleném níže. Zobrazení bludiště je řešeno stejným způsobem jako na hlavní stránce s tím rozdílem, že není interaktivní. Slouží pouze k zobrazování jednotlivých kroků v rámci určené trasy vybraného hráče. Do budoucna by nebylo špatné vyhnout se manuálnímu překlíkávání kroků implementací řízené animace, což formát SVG umožňuje.

2.4 Testování

Testování labyrintu probíhalo za pomoci sedmi mnou vybraných hráčů po dobu jednoho měsíce. Přitom se odhalila spousta neduhů, která vyústila v pár vylepšení. Většinou šlo o zpříjemnění a zpřehlednění uživatelského rozhraní, jmenovitě:

- barevné a tvarové odlišení místností (navštívených a výchozích),
- šipky pro překlíkávání poschodí bludiště a kroků v sledování tras,
- detekce duplicitních tras v žebříčcích,
- nový systém skóre formou času,
- větší detailnost pravidel pro lepší pochopení hry.

Především však byly odhaleny 2 chyby. První při tvorbě náhodného bludiště, kdy mohlo dojít k tomu, že pro něj neexistovalo správné řešení. Toto bylo vyřešeno, chyba byla pouze v dotazu do databáze.

Druhou chybou byla nefunkčnost bludiště pro prohlížeče Internet Explorer a Edge. Z důvodu přítomnosti bugu v knihovně jQuery verze 1.7.2 (34). Kvůli tomu v těchto prohlížečích nefungovalo zobrazování vyskakovacích okýnek místností, a tudíž nebylo možné se zapojit do hry. Problém vyřešila záměna používané události na místnostech `onmouseenter` za `onmouseover`. Tím však v těchto prohlížečích vyvstal další problém, tentokrát čistě grafického rázu, kdy nesprávně fungují elementy `<marker>` přetvářející čáry na šipky. Tento problém jsem zatím nevyřešil, jelikož se jedná o chybu v implementaci IE a Edge (35). Naštěstí toto nijak nebrání v hraní.

2.5 Nasazení a údržba

Po otestování a vyladění celého systému jsem již mohl Labyrint otevřít pro všechny uživatele IF portálu. Díky předchozímu pečlivému testování nedošlo k žádným vážným komplikacím, a tak jedinou činností v rámci údržby bylo monitorování stavu řešení bludiště, a pokud došlo k nalezení optimální cesty, pak vygenerování bludiště nového.

3 Výsledky

Po ostrém startu experimentu a shromáždění prvních dat nastal čas pro zhodnocení jeho účinnosti. Jelikož neexistuje (nebo se mi nepodařilo nalézt) žádný podobný gamesourcingový projekt, nemohl jsem provést porovnání na základě metod popsanych v kapitole 1.3.4. Místo toho jsem se rozhodl pro srovnání s různými variantami mravenčích algoritmů. Tedy de facto souboj výkonu mravenců zastoupených lidmi proti těm ovládaných počítačem.

K tomu jsem si stáhl open-source program AntCol¹⁵, který nabízí 6 různých variant ACO včetně nastavitelných parametrů a možností mutace feromonů. Rovněž (a především) je možné mravence spouštět i na uživatelem definovaných grafech. Vše je zaobaleno pohodlným a poměrně přehledným GUI. Jedinou nevýhodou je nemožnost změny počtu mravenců, který je stanoven na čísle deset.

3.1 Mravenčí algoritmy

Následuje popis podob mravenčích algoritmů, které jsou v AntCol k mání, a se kterými budu porovnávat výkon svého experimentu Labyrint.

3.1.1 Ant System

Autorem je Marco Dorigo (36). Nejdříve je každé hraně grafu přiřazeno počáteční množství feromonů, které je rovno převrácené hodnotě počtu uzlů v grafu. Poté každý mravenec začne v náhodném vrcholu a projde všemi ostatními vrcholy. Z posledního se vrátí do počátečního. Mravenci si vybírají uzly dle následujícího pravidla:

$$P(c_{ij}) = \frac{t_{ij}^{\alpha} * h_{ij}^{\beta}}{\sum_{c_{ij} \in N} t_{ij}^{\alpha} * h_{ij}^{\beta}}, \forall c_{ij} \in N$$

Kde:

- N – množina všech nenavštívených hran dostupných z aktuálního vrcholu,
- c_{ij} – hrana z vrcholu i do j ,
- P – pravděpodobnost,
- t_{ij} – množství feromonu na hraně c_{ij} ,
- h_{ij} – konstanta, obvykle $h_{ij} = Q/d_{ij}$, kde d_{ij} je vzdálenost mezi vrcholy i, j a Q konstanta,
- α, β – parametry algoritmu.

Mravenec porovná $P(c_{ij})$ s náhodně vygenerovaným číslem od 0 do 1, a pokud je mu $P(c_{ij})$ větší nebo rovno, mravenec se po dané hraně okamžitě vydá. Tzn., že se mravenec nemusí vždy vydat po hraně s nejvyšším feromonem, což snižuje šanci na uvíznutí v lokálním extrému.

Poté, co všichni mravenci dorazí do počátečního bodu, dojde k přepočtu feromonů na hranách dle vztahu:

¹⁵ K dispozici na: <https://www.codeproject.com/Articles/644067/Applying-Ant-Colony-Optimization-Algorithms-to-Sol/>

$$t_{ij} = t_{ij} + \sum_{k=1}^{k=m} \Delta t_{ij}^k$$

Kde:

- m – počet mravenců,
- Δt_{ij}^k – je rovno $\frac{Q}{L_k}$ – pokud mravenec k hranou mezi uzly i, j prošel, kdy Q je konstanta a L_k délka vzdálenosti, kterou mravenec k urazil – jinak je rovno nule.

Následně je třeba část feromonů odpařit. Feromony na všech hranách v grafu se přenásobí $(1 - \rho)$, kde ρ je posledním (třetím) parametrem algoritmu.

Tímto krokem končí jedna iterace a další pokračuje novými pohyby mravenců.

3.1.2 Elitist Ant System

Pohybování mravenců a odpařování feromonů probíhá stejně jako v Ant System, ale přepočít feromonů je odlišný. Doposud nejlepší mravenec umístí na hrany své nejpovedenější trasy feromony navíc:

$$t_{ij} = t_{ij} + \sum_{k=1}^{k=m} \Delta t_{ij}^k + e * \Delta t_{ij}^{bs}$$

Kde:

- t_{ij}^{bs} – je rovno $\frac{Q}{c^{bs}}$, pokud hrana mezi i, j náleží do doposud nejlepšího řešení. e je parametrem algoritmu, doporučuje se hodnota 1 a výše.
- c^{bs} – délka globálně nejlepší trasy.

3.1.3 Ranked Ant System

Opět je zde pouze rozdíl ve způsobu rozmístění feromonů. V každé iteraci se vybere globálně nejlepší řešení a dalších $w-1$ nejlepších řešení v rámci iterace. Hrany obsažené v těchto řešeních jsou potom oferomonovány takto:

$$t_{ij} = t_{ij} + \sum_{r=1}^{r=w-1} e * (w - r) * \Delta t_{ij}^r + e * w * \Delta t_{ij}^{bs}$$

- $\Delta t_{ij}^r = \frac{Q}{L^r}$, kde Q je konstanta, L^r je délka cesty mravence r a e je parametr. Nejlepších výsledků je dosahováno pro jeho hodnoty větší nebo rovny jedné.

3.1.4 Best-Worst Ant System

Znovu se zde odlišně počítají feromony. Jsou přidány pouze na hranách patřících do globálně nejlepšího řešení.

Při odpařování dojde k dodatečnému úbytku feromonů na hranách, které jsou obsaženy v trase nejhoršího řešení iterace a zároveň nejsou v onom globálním nejlepším řešení. Tyto feromony jsou přenásobeny $(1 - \rho_w)$, kde ρ_w je parametrem algoritmu.

3.1.5 Min-Max Ant System

Tato varianta obsahuje 2 další parametry t_{min} a t_{max} . Přidání feromonů probíhá jako $\min(t_{ij} + \Delta t_{ij}^{bs}, t_{max})$, odpaření feromonů potom $\max((1 - \rho) * t_{ij}, t_{min})$.

3.1.6 Ant Colony System

Ve všech výše zmíněných algoritmech docházelo k úpravám feromonů až ve chvíli, kdy všichni mravenci ukončili svou pouť grafem. Zde je to jinak. Po každém přesunu mravence dojde na použité hraně k okamžité úpravě feromonu.

3.2 Labyrint (Human Ant System)

Algoritmus skrytý za Labyrintem jsem ze zřejmého důvodu nazval Human Ant System. Každý hráč totiž reprezentuje jednoho mravence. Hlavními rozdíly oproti výše zmíněným alternativám mravenčích algoritmů jsou tyto:

1. **O přesunu mravence rozhoduje člověk.** To je největší a nejzásadnější rozdíl. Ve hře jsou hráči sice předkládány statistické informace o dostupných hranách, ale finální rozhodnutí záleží jen a pouze na něm. Důvody k jeho rozhodnutí mohou být velice komplexní a neomezují se pouze na porovnání vypočtené pravděpodobnosti s náhodně generovaným číslem od 0 do 1, jako je tomu v případě mravenčích algoritmů.
2. **Feromony.** Feromony jako takové nejsou ve hře přítomny. Každý hráč se však může kdykoliv podívat na kteroukoliv trasu kteréhokoliv hráče a podle toho se zařídít. Dá se předpokládat, že sledovány budou hlavně ty lepší trasy, naopak ty nejhorší nebudou zajímat téměř nikoho. Tím je simulováno přidávání a odpařování feromonů. Pokud se však hráč rozhodne tímto vůbec nezabývat, nikdo mu v tom nebude bránit.
3. **Počáteční bod.** Výběr výchozího bodu závisí na úvaze hráče. Díky tomu se nemusí jednat o čistě náhodnou záležitost, jako je tomu u ostatních algoritmů.
4. **Neuvázne v lokálním extrému.** Díky herní mechanice, kdy duplicitní cesty nejsou nijak odměňovány, je dosaženo toho, že hráči se neustále snaží hledat nové varianty průchodu grafem a chtějí se vyhnout tomu, aby šli stejnou trasou jako již někdo před nimi. Tím nikdy nemůže algoritmus uváznout v lokálním extrému a zacyklit se.
5. **Paralelismus.** Mravenci (hráči) se v této podobě algoritmu pohybují po grafu naprosto nevázaně. Nemusí na sebe vzájemně čekat. A tak se může klidně stát, že jeden již graf projde desítkrát, zatímco druhý jen jednou. Tím pádem se zde nelze bavit o iteracích v rámci celé kolonie. Každý mravenec si je svým pánem. Může se rozhodnout procházet grafem až poté, co mu ostatní vyloženě „naservírují“ správnou cestu. Takovým vyčkáváním však zároveň riskuje, že jej někdo předběhne. Tím se dostává k poslednímu bodu.
6. **Motivace.** Na rozdíl od umělých mravenců, ty lidské pohání chuť vyhrát. Je to vlastnost, kterou má v různé míře každý z nás. Snaha předhlonit ostatní, stát se tím nejlepším – to nelze počítačem přesně replikovat.

3.3 Porovnání

Zde se již dostávám k popisu samotného porovnání. ACO algoritmy jsem ve všech případech spouštěl s 10 mravenci a těmito parametry:

- **Ant System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$

- **Elitist Ant System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$; $e = 1$
- **Rank-Based And System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$; $w = 6$; $e = 1$
- **Best-Worst Ant System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$; $\rho_w = 0,5$
- **Min-Max Ant System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$; $t_{min} = 0$; $t_{max} = 5$
- **Ant Colony System:** $\alpha = 1$; $\beta = 1$; $\rho = 0,1$; *lokální evaporace* 0,1

Na začátku bylo ve všech případech na všechny hrany umístěno $1/n$ feromonu, kde n je počet uzlů v grafu. Počet iterací byl 10 000, nebo dokud nebylo nalezeno správné řešení. Každá varianta algoritmu byla spuštěna pětkrát. S průměrem výsledků každé varianty jsem potom srovnával výsledky Labyrintu, potažmo algoritmu Human Ant System. Jelikož Labyrint nepracuje po iteracích, srovnání probíhalo podle počtu pohybů mravenců (Ant Moves, zkráceně AM). U případů, kde některý algoritmus zkonvergoval do lokálního minima, jsem počítal AM do chvíle, kdy došlo k uváznutí. Kvalita řešení znamená poměr ideální vzdálenosti k nejlepší nalezené.

Následuje výčet úloh pro TSP a informace, jak se s nimi který algoritmus vypořádal.

1. Burma14

Jedná se o mapu 14 měst v Barmě. Nejkratší možná vzdálenost pro TSP je 3 323 km.¹⁶

Tabulka 2 – Výsledky ACO algoritmů na problému Burma14

Algoritmus	Průměrná kvalita řešení (%)	Nejlepší řešení (při počtu AM)	Průměr AM k dosažení řešení
AS	100	3 323 (28 140)	85 932
Elitist AS	100	3 323 (7 420)	32 788
Ranked AS	100	3 323 (7 560)	10 808
Best-Worst AS	95,79	3 336 (6 020)	6 048
Min-Max AS	99,21	3 323 (11 480)	12 544
Colony AS	99,77	3 323 (252 280)	582 316
Human AS	100	3 323 (1 101)	1 101

Do hraní Labyrintu se v tomto případě zapojilo celkem 37 lidí a vyřešit TSP na tomto grafu se jim podařilo za 64 hodin, 2 minuty a 52 sekund.

2. Ulysses16

Mapa Odysseovy cesty vedoucí přes 16 míst. Ideální okružní trasa je dlouhá 6 859 km.¹⁷

Zde si hru zahrálo 8 hráčů po dobu pouhých 7 hodin, 3 minut a 52 sekund.

¹⁶ Dostupné na: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

¹⁷ Dostupné na: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

Tabulka 3 – Výsledky ACO algoritmů na problému Ulysses16

Algoritmus	Průměrná kvalita řešení (%)	Nejlepší řešení (při počtu AM)	Průměr AM k dosažení řešení
AS	100	6 859 (38 080)	196 960
Elitist AS	100	6 859 (56 640)	62 464
Ranked AS	99,85	6 859 (12 480)	14 976
Best-Worst AS	96,53	6 982 (9 600)	9 216
Min-Max AS	99,81	6 859 (10 400)	13 888
Colony AS	99,45	6 870 (455 360)	688 064
Human AS	100	6 859 (442)	442

3. Gr17

Problém se 17 městy od Groetschela. Hledaná vzdálenost činí 2 085 km.¹⁸

Tabulka 4 – Výsledky ACO algoritmů na problému Gr17

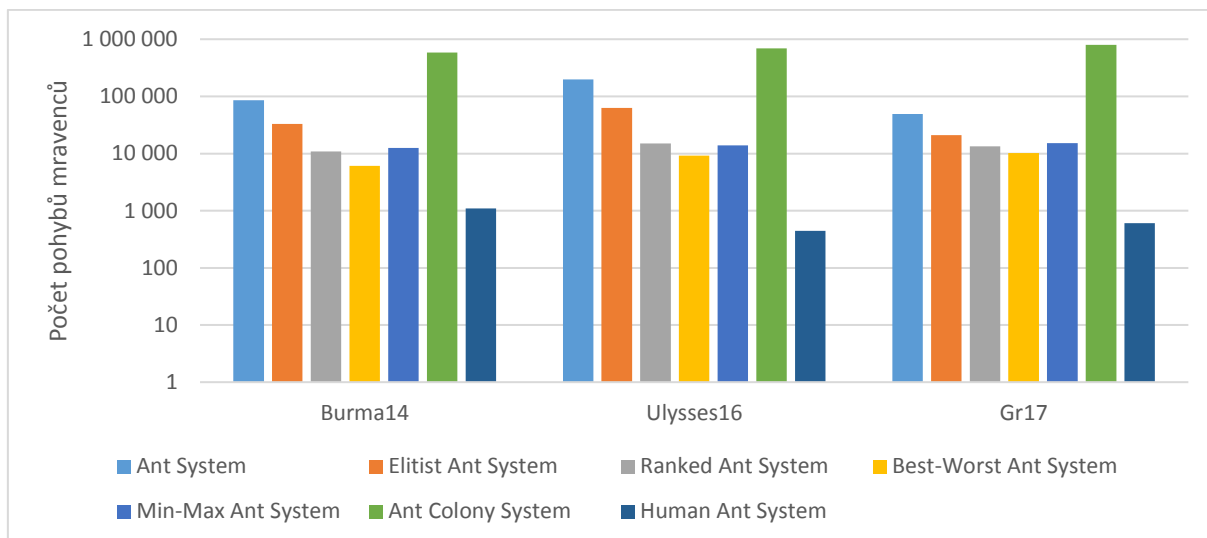
Algoritmus	Průměrná kvalita řešení (%)	Nejlepší řešení (při počtu AM)	Průměr AM k dosažení řešení
AS	100	2 085 (20 060)	49 198
Elitist AS	100	2 085 (11 560)	21 114
Ranked AS	99,19	2 085 (11 390)	13 924
Best-Worst AS	95,83	2 095 (9 350)	10 166
Min-Max AS	98,62	2 085 (14 790)	15 232
Colony AS	99,01	2 085 (107 950)	796 280
Human AS*	99,86	2 088 (603)	603

V době psaní této kapitoly zatím nebyl tento problém Labyrintem vyřešen. Do řešení se zapojilo 28 lidí, kteří během 11 dní provedli celkem 1 035 mravenčích pohybů. Dá se předpokládat, že za 1 700 000 AM (10 000 iterací s deseti mravenci), které měly v tomto případě k dispozici ostatní algoritmy, se hráčům podaří ideální cestu naleznout – a opět o mnoho rychleji.

Z výběru ACO algoritmů se pouze klasické a elitistické variantě vždy podařilo nalézt správné řešení do 10 000 iterací. Ranked, Best-Worst, Min-Max alternativy rychle konvergovaly a v 28 případech ze 45 uvázly v lokálním minimu (Best-Worst uvázl pokaždé). Ant Colony System nevykazoval konvergenci, ale ve 12 pokusech z 15 nedokázal v 10 000 iteracích dosáhnout správného řešení. Je však možné, že při dalších iteracích by jej dosaženo bylo.

Human Ant System na grafech se 14 a 16 vrcholy daleko předčil všechny ostatní a na grafu z problému Gr17 se mu podařilo nalézt velice uspokojivé řešení při mnohonásobně nižším počtu pohybů mravenců než konkurence.

¹⁸ Dostupné na: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>



Graf 6 – Srovnání ACO algoritmů z hlediska potřebného počtu pohybů mravenců (logaritmická vertikální osa)

V tuto chvíli nelze jasně říct, čím je tak markantní úspěch lidmi poháněnou ACO způsoben. Odpověď leží v datech, jež byla pořízena během hraní. Každý hráčův pohyb byl zaznamenán spolu s časovou známku a aktuální vzdáleností v danou chvíli. Analýza těchto dat však nebude jednoduchá a pro nalezení vzorců v chování, které umožňuje řešit problém obchodního cestujícího s až dvacetkrát menším počtem potřebných kroků oproti současným verzím ACO, bude dle mého názoru třeba využít i poznatky z oblasti psychologie a neurologie. Dokážu si také představit neuronovou síť, která by se na těchto datech naučila a následně produkovala cesty grafem.

Při sledování průběhu hry u každého bludiště jsem si všiml jisté podobnosti. Jakmile se objevil někdo, kdo našel cestu s o mnoho lepším skóre, než byla ta dosavadní nejlepší, spustilo to vlnu masivního zlepšování všech následujících výsledků a velice rychle se pak dosáhlo ideálního řešení. Jedná se zřejmě o důsledek principu, který jsem zmínil v kapitole 3.2 – je naší vrozenou lidskou vlastností chtít být tím nejlepším. A to je pravděpodobně nejdůležitější hnací síla celého Labyrintu.

Všechna nashromážděná data spolu s jejich statistickým zpracováním, které je výše, jsou k dispozici v příloze na CD.

Závěr

V rámci této práce jsem shrnul základní informace o gamesourcingu. Jeho kořeny, podstatu, výhody a základní paradigmaty pro jeho tvorbu. Rovněž jsem vypíchl několik současných projektů, jejich cíle a dosažené výsledky a sám si je ozkoušel.

Na základě této teoretické průpravy jsem následně vytvořil hru, která umožňuje řešit symetrickou variantu optimalizačního problému obchodního cestujícího, tedy hledání nejkratší hamiltonovské kružnice v úplném neorientovaném grafu, formou kolektivního procházení bludiště. Bylo třeba hru nadefinovat tak, aby obsahovala bariéry a principy, kterými hráče nutí fungovat na bázi podobné algoritmům optimalizace za pomoci mravenčích kolonií, jenž jsou i dnes na TSP s úspěchem aplikovány. Práce obsahuje kompletní popis celého procesu vývoje této gamesourcingové aplikace, jíž jsem nazval Labyrint, včetně všech implementačních detailů. Původní ideou bylo vytvořit samostatný program umožňující hru více hráčů, ale nakonec jsem jako výhodnější shledal využití již stávajícího online herního portálu, do něhož jsem Labyrint zakomponoval. Díky tomu jsem nemusel složitě shánět hráče ani nijak významně hru propagovat. Labyrint byl vřele přijat a dle množství hrajících hráčů, které se i nadále drží na stejné hladině, lze usoudit, že se koncept povedl a hra baví.

Výkonnost svého experimentu jsem porovnal s různými variantami ACO na třech instancích TSP. Labyrint vykázal mnohonásobně nižší počet potřebných kroků k vyřešení problému. Důvodem je zcela jistě právě ona lidská složka ve spojení se strukturou algoritmu, který skrze hru hráči prováděli. Ten je v mnoha ohledech od ACO odlišný a tyto rozdíly jsem vypsál. Hlavním faktorem je dle mého názoru nesynchronizovaný paralelismus, který v porovnávaných ACO algoritmech není.

Při hraní Labyrintu se logují data o akcích hráčů. V jejich správné analýze leží klíč k odhalení vzorců chování, které by poskytly podklady ke vzniku nového optimalizačního algoritmu pro problém obchodního cestujícího. Zde je prostor pro rozšíření této práce. Zpracování těchto dat však nebude triviální. Jednou z možných metod by mohlo být využití neuronových sítí, dost možná ale dojde i na přesah do oborů mimo tradiční informatiku jako jsou např. psychologie nebo neurologie z důvodu přítomnosti faktoru lidského uvažování, které samo o sobě umí být dost komplexní a komplikované.

Shromážděné údaje z hraní jsou k dispozici na přiloženém CD. Projekt Labyrint ale stále aktivně funguje a nadále shromažďuje cenná data, která jsem na požádání schopen kdykoliv a komukoliv dodat.

Reference

1. Kuipers, Shelley. Citizen Crowds Can Make a Big Impact on Elections. *Daily Crowdsourc*. [Online] <http://dailycrowdsourc.com/content/open-innovation/1286-crowd-leader-shelley-kuipers-citizen-crowds-can-make-a-big-impact-on-elections>.
2. Fish, LP. Island tvoří novou ústavu, on-line. *Aktuálně.cz*. [Online] 14. červen 2011. <http://zpravy.aktualne.cz/ekonomika/technika/island-tvori-novou-ustavu-on-line/r~i:article:703969/>.
3. Frojdová, Marie. Crowdsourcing. *wikisofia*. [Online] 10. leden 2015. <http://wikisofia.cz/index.php/Crowdsourcing>.
4. Handl, Jan. Crowdsourcing není novinka. *Lupa.cz*. [Online] 14. srpen 2010. <http://www.lupa.cz/clanky/crowdsourcing-neni-novinka/>.
5. „Problém zeměpisné délky“. *Online knihovna Strážné věže*. [Online] 2010. <http://wol.jw.org/cs/wol/d/r29/lp-b/102010170>.
6. Ellis, Sally. A History of Collaboration, a Future in Crowdsourcing: Positive Impacts of Cooperation on British Librarianship. *crowd consortium*. [Online] 2014. <http://www.crowdconsortium.org/wp-content/uploads/A-History-of-Collaboration-a-Future-in-Crowdsourcing-Positive-Impacts-of-Cooperation-on-British-Librarianship.pdf>.
7. Winchester, Simon. *The surgeon of Crowthorne: a tale of murder, madness and the love of words*. místo neznámé : Viking, 1998.
8. Sochorek, Radim. Oxford English Dictionary slaví 80 let. *Sochorek*. [Online] 14. říjen 2008. <http://www.sochorek.cz/cz/pr/blog/1224018334-oxford-english-dictionary-slavi-80-let.htm>.
9. 5 Famous Logo Contests - Toyota, Google, Wikipedia & More! *Design Crowd*. [Online] 19. Listopad 2010. <http://blog.designcrowd.com/article/218/5-famous-logo-contests--toyota-google-wikipedia--more>.
10. Crowdsourcing Is Not New - The History of Crowdsourcing (1714 to 2010). *DesignCrowd*. [Online] 28. Říjen 2010. <http://blog.designcrowd.com/article/202/crowdsourcing-is-not-new--the-history-of-crowdsourcing-1714-to-2010>.
11. Grier, David Alan. When Computers Were Human. *YouTube*. [Online] 6. Červen 2005. <https://www.youtube.com/watch?v=YwqltwvPnkw>.
12. Von Ahn, Luis a Dabbish, Laura. Designing Games With A Purpose. *Carnegie Mellon University*. [Online] Červenec 2008. http://www.cs.cmu.edu/~biglou/GWAP_CACM.pdf.
13. Stork, David G. The Open Mind Initiative: An internet based distributed framework for developing. *Johns Hopkins Whiting School of Engineering*. [Online] 14. Listopad 2000. <http://www.clsp.jhu.edu/events/the-open-mind-initiative-an-internet-based-distributed-framework-for-developing-david-g-stork-ricoh-silicon-valley/#.WH-RYFPhBhE>.
14. Howe, Jeff. The Rise of Crowdsourcing. *Wired*. [Online] 1. Červen 2006. <https://www.wired.com/2006/06/crowds/>.

15. Šamanová, Gabriela. Volný čas. *Centrum nezávislého průzkumu veřejného mínění*. [Online] 14. Leden 2010. http://cvvm.soc.cas.cz/media/com_form2content/documents/c1/a3718/f3/100994s_OZ100114.pdf.
16. Šafr, Jiří a Patočková, Věra. Trávení volného času v České republice ve srovnání s evropskými zeměmi. *Centrum nezávislého průzkumu veřejného mínění*. [Online] 2010. http://cvvm.soc.cas.cz/media/com_form2content/documents/c3/a4013/f11/100119s_Traveni%20volneho%20casu.pdf.
17. Internet Users. *Internet Live Stats*. [Online] 25. Leden 2017. <http://www.internetlivestats.com/internet-users/>.
18. Spil Games. State of Online Gaming Report. [Online] 2013. http://auth-83051f68-ec6c-44e0-afe5-bd8902acff57.cdn.spilcloud.com/v1/archives/1384952861.25_State_of_Gaming_2013_US_FINAL.pdf.
19. Stevens, John. Internet Stats & Facts for 2016. *Hosting Facts*. [Online] Srpen. 11 2016. <https://hostingfacts.com/internet-facts-stats-2016/>.
20. Yee, Nick. 7 Things We Learned About Primary Gaming Motivations From Over 250,000 Gamers. *Quantic Foundry*. [Online] Prosinec. 15 2016. <http://quanticfoundry.com/2016/12/15/primary-motivations/>.
21. Locke, Edwin A. a Latham, Gary P. New Directions in Goal-Setting Theory. *University of Baltimore*. [Online] 2006. <https://home.ubalt.edu/tmitch/642/Articles%20syllabus/locke%20latham%20new%20dir%20GS%20curr%20dir%20psy%20sci%202006.pdf>.
22. Astro Drone Data. *European Space Agency*. [Online] 2017. http://www.esa.int/gsp/ACT/ai/projects/astrodrone_scientific.html.
23. Templeton, Graham. Players of crowdsourcing game beat supercomputers at designing RNA molecules. *Geek.com*. [Online] 29. Leden 2014. <https://www.geek.com/news/players-of-crowdsourcing-game-beat-supercomputers-at-designing-rna-molecules-1583401/>.
24. The Science Behind Foldit. *foldit*. [Online] 5. Únor 2017. <http://fold.it/portal/info/about>.
25. Burke, Adrienne. Games That Solve Real Problems: Crowdsourcing Biochemistry. *Forbes*. [Online] 27. Říjen 2011. <http://www.forbes.com/sites/techonomy/2011/10/27/games-that-solve-real-problems-crowdsourcing-biochemistry/#695c81211990>.
26. Owens, Jo. Can the power of the public help personalise cancer treatment? *Cancer Research UK*. [Online] 1. Březen 2013. <http://scienceblog.cancerresearchuk.org/2013/03/01/can-the-power-of-the-public-help-personalise-cancer-treatment/>.
27. *Sea Hero Quest*. [Online] 6. Únor 2017. <http://www.seaheroquest.com/cs/faq>.
28. Davendra, Donald. *Traveling Salesman Problem, Theory and Applications*. místo neznámé : InTech, 2010.

29. Puget, Jean Francois. No, The TSP Isn't NP Complete. *IBM*. [Online] 29. Prosinec 2013. https://www.ibm.com/developerworks/community/blogs/jfp/entry/no_the_tsp_isn_t_np_complete?lang=en.
30. Solving the Travelling Salesman Problem Using the Ant Colony Optimization. *Ekonomski fakultet u Subotici*. [Online] 23. Září 2011. http://www.ef.uns.ac.rs/mis/archive-pdf/2011%20-%20No4/MIS2011_4_2.pdf.
31. Rad, Chloi. Minecraft sales surpass 100 million copies. *IGN*. [Online] 2. Červen 2016. <http://www.ign.com/articles/2016/06/02/minecraft-sales-surpass-100-million-copies>.
32. Fenomén Minecraft najde uplatnění i ve školách. *Novinky.cz*. [Online] Listopad. 2 2016. <https://www.novinky.cz/internet-a-pc/hry-a-herni-systemy/419424-fenomen-minecraft-najde-uplatneni-i-ve-skolach.html>.
33. Černý, Jakub. Rovinné grafy. *Základní grafové algoritmy*. [Online] <http://algoritmy.eu/zga/grafy-a-stromy/rovinne-grafy/>.
34. *jQuery*. [Online] 27. Červenec 2012. <https://bugs.jquery.com/ticket/12067>.
35. *Microsoft Connect*. [Online] 23. Září 2013. <https://connect.microsoft.com/IE/feedback/details/801938/dynamically-updated-svg-path-with-a-marker-end-does-not-update>.
36. geoyar. Applying Ant Colony Optimization Algorithms to Solve the Traveling Salesman Problem. *Code Project*. [Online] 13. Září 2013. <https://www.codeproject.com/Articles/644067/Applying-Ant-Colony-Optimization-Algorithms-to-Sol>.